This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3050571, IEEE Transactions on Knowledge and Data Engineering

Intention-aware Sequential Recommendation with Structured Intent Transition

Haoyang Li, Xin Wang, *Member, IEEE,* Ziwei Zhang, Jianxin Ma, Peng Cui, *Senior Member, IEEE,* Wenwu Zhu, *Fellow, IEEE*

Abstract—Human behaviors in recommendation systems are driven by many high-level, complex, and evolving intentions behind their decision making processes. In order to achieve better performance, it is important for recommendation systems to be aware of user intentions besides considering the historical interaction behaviors. However, user intentions are seldom fully or easily observed in practice, so that the existing works are incapable of fully tracking and modeling user intentions, not to mention using them effectively into recommendation. In this paper, we present the **I**ntention-Aware **S**equential **Rec**ommendation (**ISRec**) method, for capturing the underlying intentions of each user that may lead to her next consumption behavior and improving recommendation performance. Specifically, we first extract the intentions of the target user from sequential contexts, then take complex intent transition into account through the message-passing mechanism on an intention graph, and finally obtain the future intentions of this target user from inference on the intention graph. The sequential recommendation for a user will be made based on the predicted user intentions, offering more transparent and explainable intermediate results for each recommendation. Extensive experiments on various real-world datasets demonstrate the superiority of our method against several state-of-the-art baselines in sequential recommendation in terms of different metrics.

Index Terms—Recommendation System, Sequential Recommendation, User Intention, Intent Transition, Structured Model

1 Introduction

NOWADAYS, recommendation systems have been deeply integrated with services that provide personalized content to users, including E-commerce, social media, and search engines, etc. Many scenarios in recommendation can be modeled as a sequential recommendation problem, i.e., using historical user behaviors to recommend what this user might be interacted with in the future. For example, in online shopping systems, content providers need to generate recommendations for users based on their historical shopping logs.

There exists a rich literature in sequential recommendations [1], [2], [3], [4], [5], [6], [7]. Some early works utilize the Markov Chain (MC) to predict the next behavior of the target user through learning a probability matrix that models the relations between the current user behavior and the next [1], [2], [3], [6], [8], [9]. With the success of Deep Neural Network (DNN), many works begin to focus on developing DNN based sequential recommendation models. Recurrent Neural Network (RNN) based methods for sequential recommendation are classic examples, which aggregate all history behaviors of users via a hidden state and achieve promising performance [10]. More recently, Transfomer, based on the self-attention mechanism, is also adopted by sequential recommendation models [4], [5] to uncover the syntactic and semantic patterns between items in user history behaviors.

• H. Li, X. Wang, Z. Zhang, J. Ma, P. Cui and W. Zhu are with the Department of Computer Science and Technology in Tsinghua University, Beijing 100084, China.
E-mail: lihy18@mails.tsinghua.edu.cn, xin_wang@tsinghua.edu.cn, zw-zhang16@mails.tsinghua.edu.cn, majx13fromthu@gmail.com, cuip@tsinghua.edu.cn, wwzhu@tsinghua.edu.cn

In practice, user behavior patterns in recommendation systems are highly driven by their intentions behind. To provide better recommendations, it is important to capture user intentions besides their historic interactions. However, existing works on sequential recommendation are hard to discover the user intentions which motivate a consumption behavior and thus lack the ability to explain the reason for a particular item to be recommended to a user. Discovering and modeling user intentions poses great challenges for sequential recommendation because user intentions are seldom fully observed, nor do they always stay static and fixed in the course of time. Furthermore, users can have multiple intentions which are correlated with each other and the changing of one user intention may lead to the changes of other intentions, which makes capturing user intentions dynamically even more difficult.

To solve these challenges, in this paper, we proposed **ISRec**[1], a structured intention-aware model for sequential recommendation. Besides being more effective in recommendation accuracy, **ISRec** is able to explain why a particular item is chosen as the candidate for the next recommendation. Specifically, we first discover user intentions from their past consumption behaviors such as rating an item, writing reviews for an item, etc., then adopt an intention graph to capture the correlations among user intentions. The structured intent transition process for the target user is modeled through the message passing schema on this intention graph and the future user intention can be obtained by conducting inference on the intention graph. As such, the final recommendation can be made based on the predicted future user intentions, with the ability to explain the reason

[1]We will release the source code at publication time.

of selecting a candidate item for the next recommendation. Therefore, our proposed **ISRec** model increases the recommendation explainability by identifying the underlying user intentions that may lead to their next consumption behaviors, providing a more transparent and explainable intermediate for sequential recommendation.

We further conduct extensive experiments on several real-world datasets, showing that the proposed **ISRec** model outperforms various state-of-the-art baselines consistently in terms of different evaluation metrics such as *Hit Ratio*, *NDCG* (normalized discounted cumulative gain) and *MRR* (mean reciprocal rank). Our promising experimental results demonstrate that the **ISRec** model can identify explainable user intentions, model the structured user intent transition process, and make accurate sequential recommendations in a more explainable way.

The contributions of this paper are summarized as follows:

- We propose to utilize user intentions behind consumption behaviors to improve both the effectiveness and the explainability in sequential recommendation.
- Our proposed intention-aware sequential recommendation model (**ISRec**) is capable of identifying user intentions as well as recognizing the structured user intent transition process to provide more transparent and explainable intermediate results for sequential recommendation.
- We conduct extensive experiments on several real-world datasets, comparing the proposed **ISRec** model with various state-of-the-art approaches. Empirical experimental results demonstrate the effectiveness and the explainability of our **ISRec** model.

We review related work in Section 2, followed by a detailed formulation of our proposed Intention-Aware Sequential Recommendation (**ISRec**) model in Section 3. Section 4 presents our experimental results including quantitative comparisons, case studies, and ablation studies. Finally, we conclude our work in Section 5.

2 Related Work

In this section, we review related works on collaborative filtering, sequential recommendation, intention-aware recommendation, and structured modeling.

Collaborative Filtering. When it comes to recommendation, collaborative filtering with no doubt serves as one of the most widely adopted strategies so far. The core idea of collaborative filtering aims at learning user preferences based on their historical behaviors. Matrix factorization, one of the most famous collaborative filtering technique, factorizes the user-item interaction matrix into two low-rank matrices where each low-rank matrix represents either latent user preferences or latent item features. In addition, item similarity based methods [11], [12] estimate user preferences through directly looking at their past consumed items and calculating the similarities between the candidate items and those consumed items. The more recent deep learning based methods [13], [14], [15] achieve massive improvement by learning highly informative user preference representations. These works do not take sequential factors into account.

Sequential Recommendation. Compared with the classic recommendation methods such as collaborative filtering [16], [17], [18] or matrix factorization [19], [20], sequential recommendation targets at capturing the temporal changing patterns of user preferences. Early works on sequential recommendation typically use Markov Chains (MC) to model users' sequential patterns based on their historical behaviors. The key assumption behind this line of works is that the next item users may consume solely depends on their last consumed item (i.e., first-order MC) or last several consumed items (i.e., high-order MC) [1], [3], [6], [9]. The huge success of Deep Neural Networks (DNN) has motivated the applications of deep models in sequential recommendation as well [4], [5], [6]. One line of works is based on RNN and its variants, which seeks to encode user history behaviors into latent representations. In particular, Hidasi *et al.* [21] employ Gated Recurrent Units (GRUs) to capture the sequences of user behaviors for session-based recommendation, and they later propose an improved version [22] with a different loss function. Liu *et al.* [7] and others [23], [24] study the problem of sequential recommendation with the contextual information taken into accounts. In addition, unidirectional [4] and bidirectional [5] self-attention mechanisms are also utilized to capture sequential patterns of user behaviors, which achieve state-of-the-art performance on sequential recommendation. However, these methods merely focus on modeling the relations between the history behaviors of the target user and her next behavior, lacking the ability to capture user intentions hidden in the behaviors. We argue it is the user intentions that drive users to conduct certain behaviors and therefore existing methods suffer from being unable to understand why the target user conducts her next behavior.

Intention-aware Recommendation. More recently, various intention-aware recommendation literatures that consider intentions in users' behavior modeling are proposed. Zhu *et al.* [25] use the category of items in users' behaviors to represent intentions directly. This method is simple and provides an intuitive way to define user intentions. Chen *et al.* [26] adopt attention mechanism to capture users' category-wise intention, which is denoted as a pair of action type and item category. In [27], a neural intention-driven method is proposed to model the heterogeneous intentions behind users' complex behaviors. Wang *et al.* [28] focus on some limitations of classical Collaborative Filtering methods, and try to disentangle the representations of users and items under different intentions. Tanjim *et al.* [29] utilize self-attention mechanism to find similarities in user behaviors and temporal convolutional network to capture users intentions. However, they pay little attention to modeling the relations between user intentions especially when users have multiple intentions affecting users' behaviors. They also ignore structured user intent transition which can provide a strong inductive bias for sequential recommendation.

Structured Modeling. The ability to understand structured relationships in raw sensory data is an important component of human cognition [30] and graphs are a natural representation to model such structured relationships. Thanks to the rapid development of Graph Neural Network (GNN), there are more and more research works focusing on structure modeling [31], [32], which generally aim to model the relationships and dynamics among nodes in graphs. By studying the structured relations behind the observed data, these models can not only improve their predictive performance but also simulate the cognitive process of human decision making. The majority of the existing works on utilizing graphs to simulate human cognitive process belong to the field of physical systems and computer vision. To overcome the limitations of models based on low-level pixel reconstruction, Kipf *et al.* [30] model the state transition of high-level objects in physical systems and Kossen *et al.* [33] explicitly reason about the relationships between objects in videos over a graph structure. However, utilizing the graph structure to identify user intentions and infer their relationships for providing better recommendations is largely unexplored in sequential recommendation. We note that there also exist several works mapping items to nodes/entities in knowledge graphs and utilizing the extra information provided by the knowledge graphs to enhance recommendation [34], [35]. These works follow a different problem setting and are therefore orthogonal to our problem in this paper.

3 Method

In this section, we first introduce the problem formulation and then present the proposed **ISRec** model in detail. Notations in this paper are summarized in Table 1.

3.1 Problem Formulation

In this paper, we consider a sequential recommendation problem where $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ denotes the set of users and $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ represents the set of items. A user behavior dataset consists of the interactions between these $|\mathcal{U}|$ users and $|\mathcal{V}|$ items. For each user $u \in \mathcal{U}$, the interaction sequence sorted in the chronological order is denoted as $\mathcal{S}_u = \left[v_1^{(u)}, v_2^{(u)}, ..., v_{|\mathcal{S}_u|}^{(u)}\right]$, in which $v_t^{(u)} \in \mathcal{V}$ is the item that user u interacted at time index t. Specifically, similar to [1], [4], [5], [6], the time index t in $v_t^{(u)}$ denotes the order in which an action occurs in $\mathcal{S}_u$ with larger t indicating a more recent interaction, and we do not consider the absolute timestamp as in temporal recommendations [10], [36].

In addition to the interaction sequence, we also consider available description information of items, e.g., item titles, categories, reviews, etc. For each item, we extract keywords from all description information and refer to these extracted keywords as *concepts*. These concepts indicate the possible intentions of users while interacting with the corresponding items and provide the source of explainability. We use an item-concept matrix $\boldsymbol{E} = [e_{i,k}, 1 \le i \le |\mathcal{V}|, 1 \le k \le K]$ to denote relations between items and concepts, where $e_{i,k} = 1$ if concept k appears in the description information of item i, $e_{i,k} = 0$ otherwise, and K is the number of concepts. In our method, the user intention is defined as a subset of all possible K concepts, denoted as a multi-hot intention vector $\boldsymbol{m}_t = [m_{t,1}, m_{t,2}, ..., m_{t,K}] \in \{0,1\}^K$. Namely, the user intentions at time index t consist of the concept k if $m_{t,k} = 1$. The intention graph is defined as a graph representing the relations between the K concepts, which consists of concept-relation-concept triples. The intention transition is defined as predicting the intentions at the next time index, which are correlated with the intentions now, conditioned on the intention graph.

TABLE 1: Notations used in this paper.

Notation	Description
$\mathcal{U}, \mathcal{V}$	user and item set
$\mathcal{S}_u$	interaction item sequence of user u
T	maximum sequence length
K	number of total concepts
λ	number of activated concepts
$\boldsymbol{E} \in \{0,1\}^{\|\mathcal{V}\| \times K}$	item-concept matrix
$d, d' \in \mathbb{N}$	latent vector dimensionality
$\boldsymbol{V} \in \mathbb{R}^{\|\mathcal{V}\| \times d}$	item embedding matrix
$\boldsymbol{C} \in \mathbb{R}^{K \times d}$	concept embedding matrix
$\boldsymbol{P} \in \mathbb{R}^{T \times d}$	positional embedding matrix
t	index of the time
$\boldsymbol{m}_t \in \mathbb{R}^K$	intention vector
$\boldsymbol{x}_t \in \mathbb{R}^d$	representation of the behavior sequence
$\boldsymbol{Z}_t \in \mathbb{R}^{K \times d'}$	intention feature matrix

Given all this information, the sequential recommendation problem can be formalized as to predict the probability over all items for every user $u \in \mathcal{U}$ at time index $t = |\mathcal{S}_u| + 1$:

$$p\left(v_{|\mathcal{S}_u|+1}^{(u)} | \mathcal{S}_u\right).$$

3.2 Model Framework

The framework of **ISRec** is shown in Fig. 1. **ISRec** consists of the following 4 modules: (1) Transformer-based Encoder: we use a two-layer transformer to encode the item sequence. As the core of the transformer, the self-attention mechanism can capture the dependencies between items in the behavior sequence. (2) Intent extraction: we extract the intentions of users from the representation of the item sequence. (3) Structured intent transition: we infer the possible user intentions at the next time index using a structured transition. (4) Intent decoder: based on the intents identified in the last module, the intent decoder predicts which item out of $\mathcal{V}$ is mostly likely to be interacted by the user. We elaborate the details of the 4 modules in the following subsections.

3.3 Transformer-based Encoder

The transformer-based encoder further consists of two submodules: the embedding submodule and the self-attention submodule.

Embedding Submodule. To represent an item sequence, we first construct an item embedding matrix $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_{|\mathcal{V}|}] \in \mathbb{R}^{|\mathcal{V}| \times d}$, where each item $v_i \in \mathcal{V}$ is represented as a d dimensional vector $\boldsymbol{v}_i$, and a concept embedding matrix $\boldsymbol{C} = [\boldsymbol{c}_1, ..., \boldsymbol{c}_K] \in \mathbb{R}^{K \times d}$, where each concept is also represented as a d dimensional vector $\boldsymbol{c}_i$. To encode the position of items in the sequence, we adopt an additional positional embedding $\boldsymbol{P} = [\boldsymbol{p}_1, ..., \boldsymbol{p}_T] \in \mathbb{R}^{T \times d}$, where $\boldsymbol{p}_i$ represents the embedding of position i, and T is a preset maximum sequence length. The representation of an element in the behavior sequence is obtained as:

$$\boldsymbol{h}_i = \boldsymbol{v}_i + \boldsymbol{p}_i + \sum_{e_{i,j}=1} \boldsymbol{c}_j, \tag{1}$$

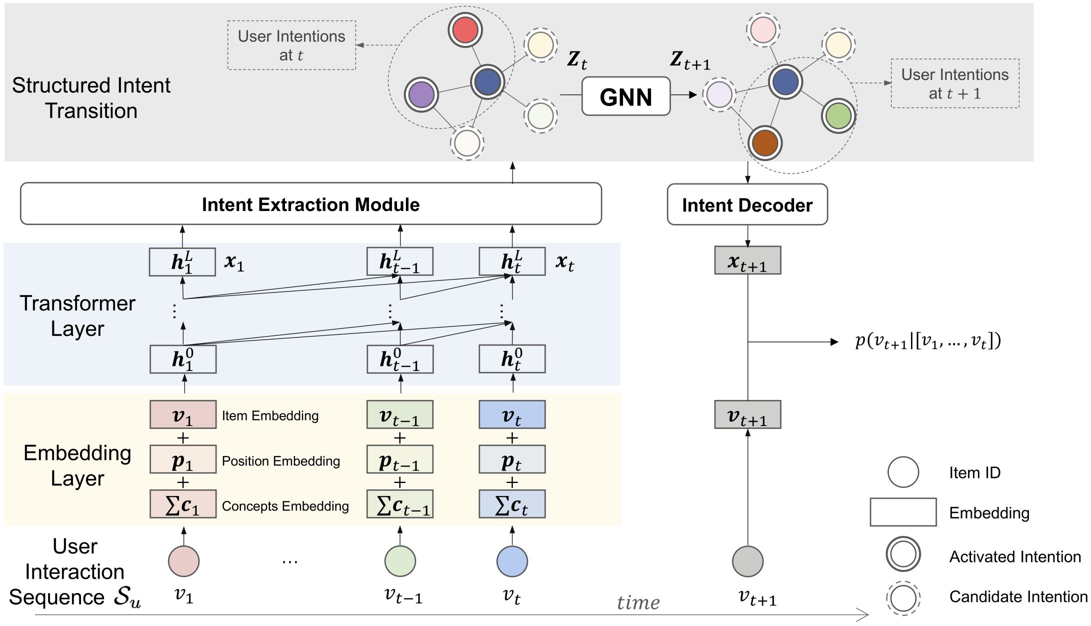


Fig. 1: **ISRec** Model Framework. After passing the user interaction sequence to a Transformer-based encoder, the keys of **ISRec** are intent-aware modules which include an intent extraction module and a structured intent transition module. Then, an intent decoder module output recommendation results using the identified user intents.

i.e., we sum the item embedding, the concepts embedding corresponding to the item, and the positional embedding. All embedding vectors are parameters that can be learned during training.

After the embedding submodule, we transform the input user behavior sequence $\mathcal{S}_u$ into its hidden representations as follows:

$$\boldsymbol{H}^0 = [\boldsymbol{h}_1^0, \boldsymbol{h}_2^0, ..., \boldsymbol{h}_T^0]. \tag{2}$$

Self-attention Submodule. We adopt the self-attention mechanism to capture the dependencies among different items within a behavior sequence. One layer in the self-attention submodule can be formulated as follows:

$$\boldsymbol{S}^l = \mathrm{SA}(\boldsymbol{H}^l) = \mathrm{Attention}(\boldsymbol{H}^l \boldsymbol{W}_Q^l, \boldsymbol{H}^l \boldsymbol{W}_K^l, \boldsymbol{H}^l \boldsymbol{W}_V^l), \tag{3}$$

$$\boldsymbol{H}^{l+1} = \mathrm{FFN}(\boldsymbol{S}^l) = \mathrm{ReLU}(\boldsymbol{S}^l \boldsymbol{W}_1^l + \boldsymbol{b}_1^l)\boldsymbol{W}_2^l + \boldsymbol{b}_2^l, \tag{4}$$

where $\boldsymbol{W}_Q^l, \boldsymbol{W}_K^l, \boldsymbol{W}_V^l \in \mathbb{R}^{d\times d}$ are parameters for queries, keys, values in the l^{th} attention layer and $\boldsymbol{W}_1^l, \boldsymbol{W}_2^l \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_1^l, \boldsymbol{b}_2^l \in \mathbb{R}^d$ are parameters in the l^{th} feed-forward network. The queries, keys, and values come from the same place, i.e., the input sequence. The meaning of queries, keys, and values is the sequence embedding. Intuitively, the attention layer learns to assign different attention weights to capture the complex relations among items in the behavior sequence[2] and the position-wise feed-forward network endows the model with nonlinearities and capture the interactions among different dimensionalities. We also apply dropout, residual connection, and layer normalization at each layer, similar to standard Transformer.

[2] To prevent data leakage, we only consider the attention between Query i and Key j if $j \leq i$, i.e., only considering attentions of items interacted ahead of time.

We denote the outputs of L such layers as $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_T] = \boldsymbol{H}^L$, which are used in subsequent modules. Note that $\boldsymbol{x}_t$ has integrated all sequential information before the time index t.

3.4 Intent Extraction

Here we explicitly extract explainable user intents from the encoded sequence hidden representations $\boldsymbol{X}$. Note that the intents are changing and not static with respect to the time index t.

More specifically, for each time index $1 \leq t \leq T$, we aim to infer an intention vector $\boldsymbol{m}_t = [m_{t,1}, m_{t,2}, ..., m_{t,K}]$, where $m_{t,k} = 1$ indicates that concept k belongs to the user intentions appearing in the behavior sequence represented as $\boldsymbol{x}_t$, and $m_{t,k} = 0$ otherwise. One straightforward approach to learn $\boldsymbol{m}_t$ is directly treating $\boldsymbol{m}_t$ as a parameter to be optimized. However, it will lead to over-parameterization and cause efficiency burdens since we need to learn a K dimensional intention vector for each user at each time index. As an alternative, recall that we have introduced an embedding vector $\boldsymbol{c}_i$ for each concept in the Transformer-based Encoder. We adopt the similarity between the sequence representation and concept embeddings as the probability of activating the concepts. Then, $\boldsymbol{m}_t$ can be drawn from the following categorical distribution:

$$\boldsymbol{m}_t \sim \mathrm{Categorical}(\mathrm{Softmax}(s_{t,1}, s_{t,2}, ..., s_{t,K})), \tag{5}$$

where $s_{t,k}$ denotes the similarity of the sequence representation $\boldsymbol{x}_t$ and the concept embedding $\boldsymbol{c}_k$. We adopt the Gumbel-Softmax estimator to estimate the categorical distribution, which is non-differentiable when trained using standard back-propagation. In choosing similarities, a common choice, the inner product similarity, will result in the mode collapse problem, i.e., only concepts with a large norm will be activated. To prevent such a degenerated case, we adopt the cosine similarity between two vectors, i.e.,

$$s_{t,k} = \frac{\boldsymbol{x}_t \cdot \boldsymbol{c}_k}{\|\boldsymbol{x}_t\|_2 \, \|\boldsymbol{c}_k\|_2}, \tag{6}$$

where $\cdot$ is the dot product and $\|z\|_2$ is the norm of vector $\boldsymbol{z}$.

3.5 Structured Intent Transition

Next, we conduct intent transitions using the extracted intention vector. However, we cannot directly transit $\boldsymbol{m}_t$ because of two reasons. Firstly, $\boldsymbol{m}_t$ is learned by using common concept embeddings and thus not personalized. Even if two users have similar intentions at time index t, their transition patterns may be different, leading to different intentions at time index $t+1$. Secondly, the intention vector $\boldsymbol{m}_t$ is discrete and contains a single number for each intention, which makes the subsequent optimization challenging.

To solve these challenges, we first learn a personalized intent feature matrix using the sequence representation $\boldsymbol{x}_t$ and the intention vector $\boldsymbol{m}_t$. Specifically, denote the intent feature matrix as

$$\boldsymbol{Z}_t = [\boldsymbol{z}_{t,1}, ..., \boldsymbol{z}_{t,K}] \in \mathbb{R}^{K \times d'}, \tag{7}$$

where d' is the dimensionality and $\boldsymbol{z}_{t,k}$ is the feature vector for intent k calculated as:

$$\boldsymbol{z}_{t,k} = m_{t,k} \mathrm{MLP}_k(\boldsymbol{x}_t), \tag{8}$$

i.e., we learn a separate MLP for each concept to transform the sequence representation into an intent feature, and only activated concepts have non-zero elements. Then, we can use $\boldsymbol{Z}_t$ for intent transition because it is both personalized and continuous.

To model the relations between different intentions, we adopt a graph $\mathcal{G}$ with the adjacent matrix denoted as $\boldsymbol{A} \in \mathbb{R}^{K \times K}$, where $\boldsymbol{A}_{i,j}$ indicates the relations between concept i and concept j. In this paper, we construct $\boldsymbol{A}$ based on the publicly available concept graph (i.e., ConceptNet[3]). $\boldsymbol{A}_{i,j} = 1$ if concept i and j have semantic relations in ConceptNet, and $\boldsymbol{A}_{i,j} = 0$ otherwise. Our method can also be extended to other available concept relations or learning the relation.

We adopt the message-passing framework [37] to model the transition of intents on the concept graph:

$$\boldsymbol{Z}_{t+1} = \mathcal{F}(\boldsymbol{Z}_t, \mathbf{A}), \tag{9}$$

where $\mathcal{F}(\cdot)$ is the message-passing function. Specifically, we adopt Graph Convolutional Network (GCN) [38], a simple yet effective message-passing architecture, where the l^{th} GCN layer is:

$$\boldsymbol{H}_{\mathcal{G}}^{l+1} = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}_{\mathcal{G}}^{l} \boldsymbol{W}^l), \tag{10}$$

where $\boldsymbol{H}_{\mathcal{G}}^{l}$ is node representations in the l^{th} layer, $\boldsymbol{W}^l$ is a learnable weight matrix, σ is a non-linear activation function such as ReLU, $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{I}$ is the identity matrix, and $\hat{\boldsymbol{D}}$ is a diagonal degree matrix with $\hat{\boldsymbol{D}}_{i,i} = \sum_j \hat{\boldsymbol{A}}_{i,j}$. Intuitively, GCNs pass the node features to their neighborhoods in each layer, thus modeling the relations between different nodes, i.e., concepts.

[3] http://conceptnet.io/

The intent transition process can be modeled as taking the intent feature matrix as the inputs of GCN, i.e., $\boldsymbol{H}_{\mathcal{G}}^0 = \boldsymbol{Z}_t$, and taking the node representations after L GCN layers as the output of future intents, i.e., $\boldsymbol{Z}_{t+1} = \boldsymbol{H}_{\mathcal{G}}^L$. Then, we obtain the new intent vector $\boldsymbol{m}_{t+1}$ by considering the norm of the corresponding intent feature vector, i.e., $m_{t+1,k} = 1$ if and only if $\|\boldsymbol{z}_{t+1,k}\|_2 \geq g(\{\|\boldsymbol{z}_{t+1,k}\|_2, 1 \leq k \leq K\})$, where g is an operator that outputs the λ-th largest value of the input. This guarantees that the number of activated concepts, i.e., λ, that remains the same in the course of time, i.e., $\sum_k m_{t,k} = \sum_k m_{t+1,k}$.

3.6 Intent Decoder

After obtaining the future intent features $\boldsymbol{Z}_{t+1}$ and the future intent vector $\boldsymbol{m}_{t+1}$, we need to make recommendations on the next item. We adopt a decoder as follows:

$$\boldsymbol{x}_{t+1} = \sum_{k=1}^{K} m_{t+1,k} \mathrm{MLP}'_k(\boldsymbol{z}_{t+1,k}). \tag{11}$$

Eq. (11) can be considered as a reverse process of Eq. (8) to decode the intent features into a sequence representation.

Then we calculate the similarity of the sequence representation with the item embedding vector to obtain a recommendation probability:

$$p(v_{t+1}|[v_1, v_2, ..., v_t]) = \mathrm{Softmax}(\boldsymbol{x}_{t+1} \boldsymbol{V}^T) \tag{12}$$

3.7 Objective Function and Optimization

Following the conventional training methods of sequential recommendation, we train the model by predicting the next item for each position in the input sequence. i.e., predicting v_{t+1} given the input sequence $[v_1, v_2, ..., v_t]$. We adopt the negative log-likelihood as the objective function and take the average of all users, i.e.,

$$\mathcal{L}_u = \frac{1}{|\mathcal{S}^{(u)}|} \sum_{v_{t+1} \in \mathcal{S}^{(u)}} -\log p(v_{t+1}|[v_1, v_2, ..., v_t]), \tag{13}$$

$$\mathcal{L} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathcal{L}_u + \alpha ||\Theta||_2^2, \tag{14}$$

where α denotes the regulation coefficient and Θ denotes all model parameters. It is easy to see that all modules of **ISRec** are differentiable and thus the model can be trained end-to-end using back-propagation. The training procedure of our method is listed in Appendix A.

3.8 Time Complexity Analysis

Here, we analyze the time complexity of the proposed method, given the user interaction item sequence with the length n. The time cost mainly comes from the following three parts, namely the Transformer layer, the Multi-layer Perceptron (MLP), and the Graph Convolutional Network (GCN). For the Transformer-based encoder, the complexity is $O(n^2 d + nd^2)$ from the self-attention and the feedforward network. The dominant term is $O(n^2 d)$ due to the self-attention, where d is the dimensionality of item embedding. Moreover, the MLP in our method has a computational complexity $O(nKdd')$, where K is the constant number of total concepts, and d' is the feature dimensionality of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3050571, IEEE Transactions on Knowledge and Data Engineering

intents. For GCN in the structured intent transition, the computational complexity is $O(\lambda^2)$, where λ represents the number of activated intentions (nodes) in the concept graph $\mathcal{G}$ and has a small value in our experiments (more details in Section 4). So the overall training complexity of our proposed method is $O(n^2d+nKdd'+\lambda^2)$. The scalability concern about our proposed method is that its computational complexity is quadratic with the input sequence length n due to the self-attention mechanism. Fortunately, a convenient property of **ISRec** is that the self-attention computation can be effectively parallelized, which is amenable to GPU acceleration.

3.9 Discussion

To provide more insights of our proposed method **ISRec**, we analyze the relationship between **ISRec** and other existing sequential recommendation methods.

Markov Chains (MC) based methods. There are many works on sequential recommendation adopting Markov Chains (MC), which can be typically divided into two types, namely first-order MC based methods (e.g., FPMC [1], TransRec [2], etc.) and high-order MC based methods (e.g., Fossil [9], Caser [6], etc.). However, these methods only capture local sequential patterns, and can not scale well with the order that is generally small. Besides, the order of MC needs to be specified in advance that is an impactive hyperparameter. Compared with these methods, our **ISRec** is conditioned on previous T items, and is able to deal with hundreds of historical interacted items empirically (more details in section 4). Due to the attention mechanism, **ISRec** can adaptively attend on informative items of input sequence instead of focusing on the last few items.

RNN based methods. RNN-based methods are recent representative works for modeling sequence, including GRU4Rec [21], GRU4Rec$^+$ [22], etc. However, these methods have a high dependency on time steps. The behavior on time step t has to wait for the results until time step $t-1$. Compared with our method, they can not be effectively parallelized using GPU.

Transformer based methods. Transformer based methods are also representative works recently. SASRec [4] adopts transformer to predict the next item for each position in a sequence. BERT4Rec [5] predicts the masked items in the sequence using Cloze objective. These methods make full use of self-attention to capture the item relations between user sequence behaviors but are incapable of capturing user intentions hidden in the behaviors. We argue that the user intentions play an important role in driving users to conduct certain behaviors. Besides, our method can be treated as a generalization of these methods. If we do not extract user intentions from behavior sequence (by removing the intent extraction module) or conduct intent transition (by removing structured intent transition module), our **ISRec** method can degenerate to the transformer based methods. In section 4, we show the significance of capturing the user intentions and structured intent transitions with ablation study.

4 EXPERIMENTS

In this section, we evaluate our proposed method through experiments. We aim to answer the following three questions:

- **Q1:** How does **ISRec** perform compared with other state-of-the-art sequential recommendation methods?
- **Q2:** Can **ISRec** identify explainable user intents and model the structured intent transition accurately?
- **Q3:** Is the intent extraction and structured intent transition module helpful in **ISRec**?

4.1 Datasets

We compare **ISRec** with baselines on five publicly available datasets from four real world applications.

- **Amazon [39][4]:** This dataset contains a large number of product reviews from $Amazon.com$ and is split into multiple datasets according to the top-level product categories. In our experiments, we choose the "Beauty" category dataset. Besides interaction records, we also extract the concepts of items from two fields (i.e., "product title" and "review text") in reviews data.
- **Steam [4][5]:** This dataset contains rich English reviews, crawled from $Steam$, a popular online video game platform. Also, we extract interaction records and concepts of items from two fields, i.e., "app name" and "review text" in reviews.
- **Epinions [40][6]:** This dataset is collected from a popular online consumer review website $Epinions.com$. It contains rating scores and review texts of users on the website, and spans more than a decade, from January 2001 to November 2013. We extract interaction records from rating scores and concepts of items from "item title" and "review text".
- **MovieLens [41][7]:** This dataset is about movie rating and has been widely used to evaluate recommendation algorithms. We use two versions, i.e., ML-1m and ML-20m, containing 1 million and 20 million rating records, respectively. We extract interaction records from rating data and concepts of each movie from "movie name", and "genre" for ML-1m and "tag" for ML-20m.

We follow the preprocessing procedure in [1], [4], [5], [6] as follows. First, we convert all reviews (for Amazon, Steam, and Epinions) or numeric ratings (for MovieLens) to implicit feedback of 1 (i.e., the user interacted with the item). Then we group the interaction records by users and build the interaction sequence sorted according to the timestamps for each user. We remove all users and items if they have fewer than 5 records. The statistics of the preprocessed datasets is summarized in Table 3, where "#Users" is the number of users, "#Items" is the number of items, and "#Interactions" means the number of interactions between users and items in each dataset. "Avg.length" denotes the average interaction sequence length of users, and "Density" is a common metric to describe how dense the user item interaction is. These datasets come from different domains and have diverse statistics.

We further obtain the concepts of items from the available meta-data, i.e., the descriptions of items. For Amazon,

[4]http://jmcauley.ucsd.edu/data/amazon/
[5]https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data
[6]https://cseweb.ucsd.edu/~jmcauley/datasets.html#social_data
[7]https://grouplens.org/datasets/movielens/

TABLE 2: Overall performance comparison of **ISRec** and baselines. In each row, the boldfaced score denotes the best result and the underlined score represents the second-best result. Our **ISRec** outperforms all the baselines consistently in all evaluation metrics on different datasets. The relative improvements of **ISRec** over the second-best result are shown in the last column.

Datasets	Metric	PopRec	BPR-MF	NCF	FPMC	GRU4Rec	GRU4Rec$^+$	DGCF	Caser	SASRec	BERT4Rec	ISRec	Improv.
Beauty	HR@1	0.0077	0.0415	0.0407	0.0435	0.0402	0.0551	0.0626	0.0475	0.0906	0.0953	**0.1233**	29.38%
	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1781	0.1835	0.1625	0.1934	0.2207	**0.2734**	23.88%
	HR@10	0.0762	0.1992	0.2142	0.2401	0.2343	0.2654	0.2778	0.2590	0.2653	0.3025	**0.3594**	18.81%
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1172	0.1241	0.1050	0.1436	0.1599	**0.2020**	26.33%
	NDCG@10	0.0349	0.1064	0.1124	0.1211	0.1074	0.1453	0.1543	0.1360	0.1633	0.1862	**0.2296**	23.31%
	MRR	0.0437	0.1006	0.1043	0.1056	0.1023	0.1299	0.1381	0.1205	0.1536	0.1701	**0.2081**	22.34%
Steam	HR@1	0.0159	0.0314	0.0246	0.0358	0.0574	0.0812	0.0564	0.0495	0.0885	0.0957	**0.1450**	51.52%
	HR@5	0.0805	0.1177	0.1203	0.1517	0.2171	0.2391	0.1825	0.1766	0.2559	0.2710	**0.3622**	33.65%
	HR@10	0.1389	0.1993	0.2169	0.2551	0.3313	0.3594	0.2934	0.2870	0.3783	0.4013	**0.5072**	26.39%
	NDCG@5	0.0477	0.0744	0.0717	0.0945	0.1370	0.1613	0.1392	0.1131	0.1727	0.1842	**0.2570**	39.52%
	NDCG@10	0.0665	0.1005	0.1026	0.1283	0.1802	0.2053	0.1717	0.1484	0.2147	0.2261	**0.3036**	34.28%
	MRR	0.0669	0.0942	0.0932	0.1139	0.1420	0.1757	0.1400	0.1305	0.1874	0.1949	**0.2612**	34.02%
Epinions	HR@1	0.0075	0.0151	0.0155	0.0162	0.0169	0.0176	0.0188	0.0164	0.0217	0.0220	**0.0282**	28.18%
	HR@5	0.0339	0.0472	0.0538	0.0578	0.0629	0.0737	0.0736	0.0733	0.0822	0.0866	**0.1129**	30.37%
	HR@10	0.0831	0.1005	0.0975	0.1083	0.1280	0.1380	0.1353	0.1351	0.1358	0.1462	**0.1949**	33.31%
	NDCG@5	0.0206	0.0316	0.0338	0.0373	0.0431	0.0456	0.0491	0.0444	0.0530	0.0534	**0.0699**	30.90%
	NDCG@10	0.0358	0.0464	0.0474	0.0512	0.0565	0.0657	0.0656	0.0642	0.0701	0.0724	**0.0962**	32.87%
	MRR	0.0430	0.0540	0.0543	0.0546	0.0681	0.0700	0.0693	0.0668	0.0699	0.0705	**0.0885**	25.53%
ML-1m	HR@1	0.0141	0.0914	0.0397	0.1386	0.1583	0.2092	0.1770	0.2194	0.2351	0.2863	**0.3184**	11.21%
	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5103	0.4485	0.5353	0.5434	0.5876	**0.6262**	6.57%
	HR@10	0.1358	0.4301	0.3477	0.5946	0.6207	0.6351	0.6032	0.6692	0.6629	0.6970	**0.7363**	5.64%
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3705	0.3162	0.3832	0.3980	0.4454	**0.4831**	8.46%
	NDCG@10	0.0621	0.2365	0.1640	0.3439	0.3627	0.4064	0.3660	0.4268	0.4368	0.4818	**0.5189**	7.70%
	MRR	0.0627	0.2009	0.1358	0.2891	0.3041	0.3462	0.3105	0.3648	0.3790	0.4254	**0.4589**	7.87%
ML-20m	HR@1	0.0221	0.0553	0.0231	0.1079	0.1459	0.2021	0.1760	0.1232	0.2544	0.3440	**0.3505**	1.89%
	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.5118	0.4361	0.3804	0.5727	0.6323	**0.6484**	2.55%
	HR@10	0.1378	0.3538	0.2922	0.5201	0.5844	0.6524	0.6252	0.5427	0.7136	0.7473	**0.7689**	2.89%
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.3630	0.3267	0.2538	0.4208	0.4967	**0.5024**	1.15%
	NDCG@10	0.0695	0.1786	0.1271	0.2895	0.3637	0.4087	0.3809	0.3062	0.4665	0.5340	**0.5401**	1.14%
	MRR	0.0709	0.1503	0.1072	0.2273	0.2967	0.3476	0.3278	0.2529	0.4026	0.4785	**0.4841**	1.17%

TABLE 3: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Avg.length	Density
Beauty	40,226	54,542	0.35m	8.8	0.02%
Steam	281,428	13,044	3.5m	12.4	0.10%
Epinions	5,015	8,335	26.9k	5.37	0.06%
ML-1m	6,040	3,416	1.0m	163.5	4.79%
ML-20m	138,493	26,744	20m	144.4	0.54%

TABLE 4: Statistics of preprocessed concepts of the datasets.

Dataset	#Concepts	#Edges	Avg.concepts/item
Beauty	592	2,791	4.45
Steam	229	472	4.49
Epinions	114	467	5.50
ML-1m	96	327	1.94
ML-20m	316	842	4.21

Steam, and Epinions dataset, we adopt the keywords in item title and review text. To reduce noises introduced by uncommon words, we only consider the keywords existing in ConceptNet [42], a widely used semantic network containing common sense concepts as well as their relationships people use in daily life. We map the n-grams in the item titles and review texts to the concepts in ConceptNet. For example, the review "I bought these athletic shoes which are comfortable." contains three concepts: athletic, shoes, and comfortable. These concepts are a subset of words that correspond to important explicit features of items and intents of users. For MovieLens, we adopt a similar approach as Amazon, Steam, and Epinions by only taking movie titles and genre/tag into account since no review information is available. For all datasets, we also filter both extremely rare concepts (occurring in less than 0.5% of reviews), domain-dependent frequent concepts, (e.g., "beautiful" in Beauty and "games" in Steam), and meaningless concepts manually. In addition, based on the chosen concepts, we build an intention graph $\mathcal{G}$ based on ConceptNet for each dataset. The graph $\mathcal{G}$ contains the relational knowledge between concepts. For example, the concept "sport" has edges with other concepts like "health", "entertainment", and "injury". The statistics of the preprocessed concepts and the filtered graph are shown in Table 4, where "#Concepts" denotes the number of concepts in each dataset, and "#Edges" denotes the number of relations. We also list the average concepts per item in the table.

4.2 Experimental Settings

4.2.1 Evaluation settings

We adopt the common leave-one-out evaluating strategy in sequential recommendation [4], [6], [43], i.e., predicting the next item in user sequence. Specifically, for each user u with interaction sequence $\mathcal{S}_u = [v_1^{(u)}, v_2^{(u)}, ..., v_{|\mathcal{S}_u|}^{(u)}]$, we hold-out $v_{|\mathcal{S}_u|}^{(u)}$ and $v_{|\mathcal{S}_u|-1}^{(u)}$ for testing and validation, respectively, and use the rest sequence for training. In addition, we follow [5] and randomly sample 100 negative items that the user does not interact with as negative items. The task is to rank these 101 items including 1 ground-truth positive item and 100 negative items.

4.2.2 Metrics

Based on the results of ranking, we evaluate all the models in terms of three commonly used criteria.

- **Hit Rate.** Hit Rate (HR) gives the percentage that recommended items contain at least one correct item interacted by the user. For each user, since we only have one ground truth item in the test set, HR@k equals to Recall@k, indicating that whether the ground-truth positive items emerge in the top-k recommended items.

$$\mathrm{HR@}k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \delta(|\mathcal{T}_u \cap \mathcal{R}_{u,k}| > 0), \tag{15}$$

where $\mathcal{T}_u$ denotes the set of testing items for user u, $\mathcal{R}_{u,k}$ is the set of top-k items recommended for user u. $\delta(x)$ is the indicator function, whose value is 1 when x is true, and 0 otherwise.

- **Normalized Discounted Cumulative Gain.** Normalized Discounted Cumulative Gain (NDCG) takes the exact position of the correctly recommended items into account.

$$\begin{aligned}\mathrm{NDCG@}k &= \frac{1}{Z}\mathrm{DCG@}k \\ &= \frac{1}{Z}\frac{1}{|\mathcal{U}|}\sum_{u \in \mathcal{U}}\sum_{i=1}^{k}\frac{\delta(r_{u,i} \in \mathcal{T}_u)}{log_2(i+1)},\end{aligned} \tag{16}$$

where $r_{u,i}$ is the k-th item recommended for user u. Z is a normalization constant, which is the maximum possible value of DCG@k.

- **Mean Reciprocal Rank.** Mean Reciprocal Rank (MRR) is the mean of reciprocal of the rank at which the ground-truth item was retrieved.

$$\mathrm{MRR} = \frac{1}{|\mathcal{U}|}\sum_{u \in \mathcal{U}}\frac{1}{rank_u}, \tag{17}$$

where $rank_u$ refers to the rank position of the ground truth item in the positive and negative items for user u.

In our experiments, k is set to 1, 5, and 10. We report the average results of these metrics across all users. For all these metrics, the higher the value, the better the performance.

4.2.3 Baselines

To verify the effectiveness of our method, we compare **ISRec** with the following recommendation baselines.

- **PopRec**: It is the simplest method that ranks all items according to their popularity, i.e., the number of existing interactions.
- **BPR-MF** [44]: It combines Bayesian personalized ranking with matrix factorization model and learns personalized rankings from implicit feedback.
- **NCF** [43]: NCF is a classical method that leverages a *Multi-Layer Perceptron* (MLP) to learn the user-item interaction function.
- **FPMC** [1]: To capture users' long-term preferences and behavior patterns, FPMC combines matrix factorization and first-order Markov chains.
- **GRU4Rec** [21]: It is a session-based recommendation method that employs GRU to characterize user behavior sequences. We treat the interaction sequence of each user as a separate session.
- **GRU4Rec$^+$** [22]: It improves **GRU4Rec** by using a new sampling strategy and an improved loss function.
- **DGCF** [28]: DGCF is an intention-aware method that considers user-item relationships at the granularity of user intentions by disentangled representations.
- **Caser** [6]: It is a unified and flexible method for capturing both general user preferences and user behavior patterns by utilizing CNN to model high-order Markov chains.
- **SASRec** [4]: It is a transformer based method that identifies which items are relevant to predict the future item from a user's behavior sequence.
- **BERT4Rec** [5]: It employs a deep bidirectional self-attention to model user behavior sequences. By adopting the Cloze objective, it predicts the random masked items in the sequence by jointly considering the left and the right context.

We do not compare against temporal recommendation methods [10], [36] because they have different settings with ours. We provide the implementation details including parameter settings in Appendix B.

4.3 Recommendation Accuracy (Q1)

We report the performance of all the methods in Table 2[8]. We make the following observations.

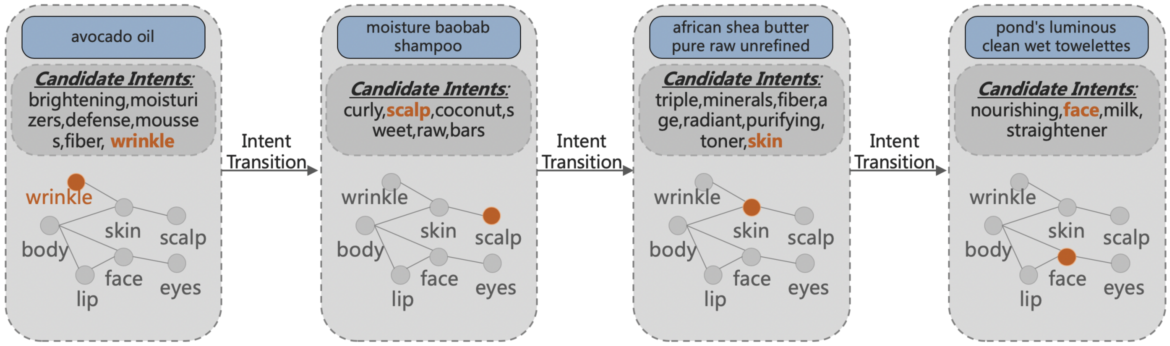
Firstly, we can see that the sequential methods (e.g., FPMC and GRU4Rec) outperform the non-sequential methods (e.g., BPR-MF and NCF) in general. The methods that only consider user actions without the sequential order, do not make full use of the sequence information and report the worse performance. Specifically, compared with BPR-MF, the main advantage of FPMC comes from modeling user historical actions with first-order Markov chains, namely considering the sequence order, so that FPMC reports better results than BPR-MF. This can verify that sequential pattern is important for improving the predictive ability for sequential recommendations.

The attention mechanism can provide reasonably large performance gains. SASRec and BERT4Rec, using a left-to-right and bidirectional self-attention respectively to model user behavior sequences, outperform the other non-attention based methods. The results are consistent with the literature [4], [5].

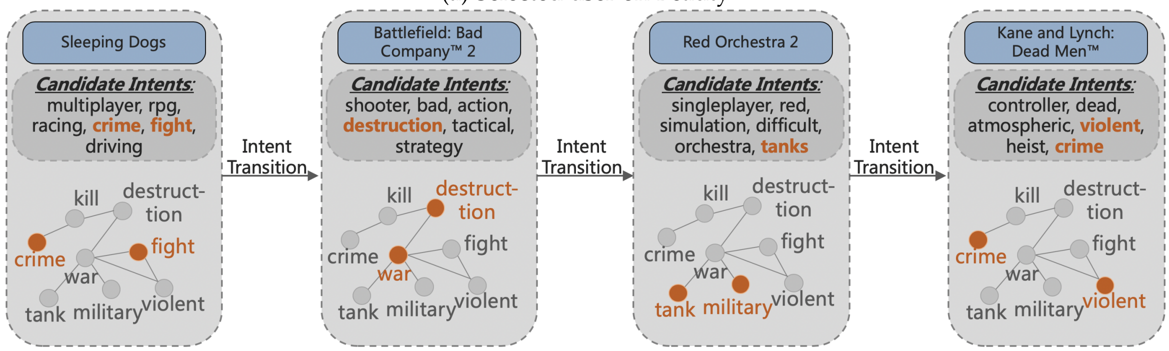
Our **ISRec** achieves the best performance on all datasets with respect to all evaluation metrics, demonstrating the superiority of our model. In general, the proposed **ISRec** model improves up to 17.41% on HR@10, 19.86% on NDCG@10, and 18.19% on MRR (on average) against the strongest baseline on all datasets. Considering the results of Steam dataset, **ISRec** achieves significant improvement, i.e., 51.52% on HR@1, 33.65% on HR@5, 26.39% on HR@10, 39.52% on NDCG@5, 34.28% on NDCG@10, and 34.02% on MRR against the strongest baseline. The fact that **ISRec** greatly

[8] In Table 2, we omit the metric NDCG@1 because it is equal to HR@1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3050571, IEEE Transactions on Knowledge and Data Engineering



(a) Selected user on Beauty



(b) Selected user on Steam

Fig. 2: Showcases of candidate intent(s) generation and activated intent(s) selection procedures for sequential recommendations made by **ISRec** on Beauty and Steam.

outperforms SASRec and BERT4Rec which adopt a similar attention module as **ISRec** but neglects the user intentions well prove the importance of modeling user intentions. **ISRec** also achieves better performance than the intention-aware method DGCF, indicating the ability of our method to model user intentions and the important roles of the structured intent transition. By identifying user intents and learning the structured intent transition, **ISRec** shows the ability to capture user preferences more effectively.

We also notice that the improvement of **ISRec** on Beauty, Steam, and Epinions datasets is more substantial than the improvement on MovieLens. **ISRec** improves over the strongest baselines $w.r.t$ NDCG@10 by 23.31% on Beauty, 34.28% on Steam, and 32.87% on Epinions but only 7.70% on ML-1m and 1.14% on ML-20m. One plausible reason is that the Beauty, Steam, and Epinions datasets are sparser, making it more difficult to make recommendations only using the co-occurrence statistics in user interaction sequences as in the baselines. **ISRec** alleviates this issue by modeling the underlying intentions and the structured transition of intentions of users and thus leading to better results.

4.4 Showcases of Intent Extraction and Structured Intent Transition (Q2)

To further illustrate the effectiveness of our intent extraction and structured intent transition process, we present the intermediate candidate intent(s) generation and activated intent(s) selection procedures for sequential recommendations made by our **ISRec** model.

Fig. 2 shows the candidate intents generation and activated intents selection procedures for two randomly selected users, one from Beauty (a) and the other from Steam (b). Each grey box represents a recommended item where the blue rectangle depicts the name of the item (e.g., avocado oil), followed by the candidate intents to be activated (e.g., brightening, moisturizers, defense, mousses, fiber, wrinkle, etc.) and the intention graph indicating the structured relationships among different intentions where the activated intentions are colored with orange (e.g, wrinkle).

We observe from Fig. 2 that the user intentions on Beauty transit from *wrinkle* through *scalp* and *skin* to *face* in the course of time, and transit gradually from *crime, fight* through *war, destruction* and *tank, military* to *crime, violent* on Steam, demonstrating the effectiveness and explainability of our structured intent transition process. **ISRec** can also learn to infer user intentions not in the candidate set, e.g., *Red Orchestra 2* is about *military*, showing its strong inference ability.

4.5 Effectiveness of Intent Extraction and Structured Intent Transition (Q3)

TABLE 5: Performance comparison of **ISRec** and variants.

	Beauty		ML-1m	
	HR@10	NDCG@10	HR@10	NDCG@10
ISRec	0.3594	0.2296	0.7363	0.5189
w/o GNN	0.3311	0.2095	0.7222	0.4978
w/o GNN&Intent	0.3092	0.1965	0.7058	0.4731
BERT4Rec + concept	0.3037	0.1886	0.6987	0.4824
SASRec + concept	0.3061	0.1845	0.6972	0.4643

To gain a deep insight on the **ISRec**, we perform ablation studies over a number of key components related

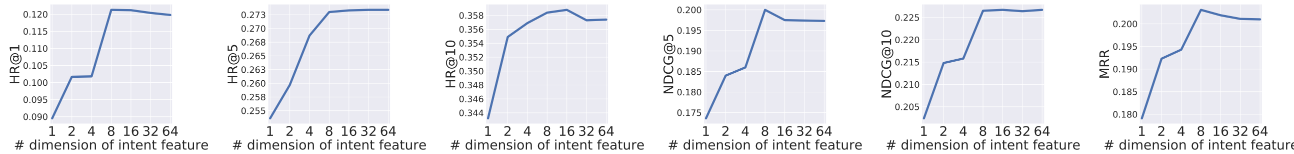


Fig. 3: Impact of different intent feature dimensionalities on model performance on Beauty.

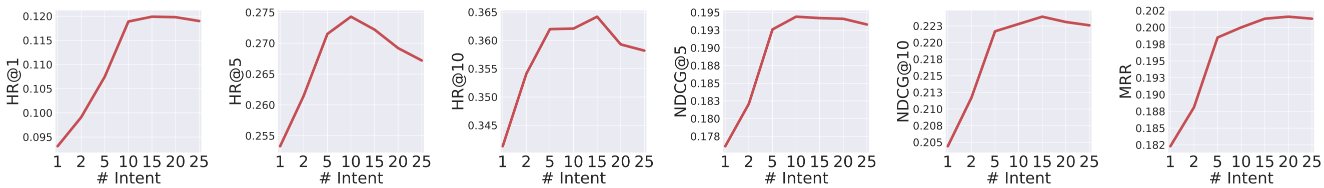


Fig. 4: Impact of different numbers of intents allowed to be activated on model performance on Beauty.

to extracting intentions and structured intent transition. We compare **ISRec** with the following two variants: one without the message-passing in Section 3.5, i.e., setting the intention feature $\boldsymbol{Z}_{t+1} = \boldsymbol{Z}_t$, and one without the message-passing nor the intention extraction module, i.e., setting $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$. We term these two variants "w/o GNN" and "w/o GNN&Intent", respectively. The results are shown in Table 5. We only report the results using the metric HR@10 and NDCG@10 on Beauty and ML-1m, while results using other metrics and datasets show a similar pattern.

- **ISRec** w/o GNN&Intent reports similar results as BERT4Rec. Since we also use a transform-based encoder, such results are consistent with our model design.
- Both intent extraction and structured intent transition modules can significantly improve the performance of **ISRec**, demonstrating the significance of accurately modeling structured transition of user intents.

We also consider incorporating available concepts for some baselines. We choose the second-best and third-best methods in Table 2, i.e., BERT4Rec and SASRec. From Table 5, we can observe the performance gain of these variants (terms as "BERT4Rec + concept" and "SASRec + concept") due to the concept information, compared with the results in Table 2. However, **ISRec** still outperforms these two variants using the same extra concept information.

4.6 Sensitivities of Hyperparameters

We also conduct experiments testing the influences of different hyperparameter settings on the performance of our **ISRec** model.

4.6.1 Impact of feature dimensionality of intents d'

Fig. 3 shows how varying the feature dimensionality of intents can affect the performance of **ISRec** on Beauty. We observe that the performance first increases with larger feature dimensions and drops after the intent feature dimensionality exceeds 8 in terms of most metrics. A larger hidden dimensionality of d' does not necessarily lead to better model performance, which is probably caused by overfitting.

4.6.2 Impact of numbers of activated intents λ

Fig. 4 presents the influences of different numbers of activated intents on the model performance. Similar to the feature dimensionality, the performance of **ISRec** first increases and then drops after a peak which occurs between 10 and 15. The results show that though setting large values for hyperparameters will increase the model capacity, it will not always lead to better results, indicating that setting hyperparameters corresponding to real user intents is helpful for **ISRec**. In our experiments, we find that uniformly setting the feature dimensionality as 8 and the number of intents as 10 leads to satisfactory performance.

4.6.3 Impact of maximum sequence length T

TABLE 6: Performance with different maximum sequence length T

	T	10	20	30	40	50
Beauty	HR@10	0.3401	0.3609	0.3608	0.3598	0.3594
	NDCG@10	0.2128	0.2304	0.2303	0.2301	0.2296

	T	10	50	100	200	300
ML-1m	HR@10	0.5873	0.7108	0.7230	0.7363	0.7360
	NDCG@10	0.3753	0.4890	0.5059	0.5189	0.5187

To verify the impact of the maximum sequence length T, we consider the different settings that T is 10, 20, 30, 40, 50 for Beauty dataset, and T is 10, 50, 100, 200, 300 for ML-1m dataset. Table 6 summarizes the performance of **ISRec** with various T. We can observe that for Beauty dataset the best performances are achieved on a small value $T = 20$, because the average sequence length of Beauty is only 8.8 (shown in Table 3). However, ML-1m dataset prefers a larger $T = 200$, because its average sequence is up to 163.5. This indicates the proper maximum sequence length T is highly dependent on the average sequence length of the dataset. Although a larger T can consider more sequence information, it will also introduce more noise. So the performances do not consistently benefit from a larger T. As the T increases, the performances of our method tend to be relatively stable, showing that **ISRec** can focus on the useful informative items and filter the noise from user interaction sequence.

5 CONCLUSIONS

In this paper, we study the intent-aware sequential recommendation problem with structured intent transition. We propose an intention-aware sequential recommendation

(**ISRec**) method which is able to discover the user intentions behind her behaviors history and model the structured user intention transition patterns. Our proposed **ISRec** model can make accurate sequential recommendations with more transparent and explainable intermediate results for each recommendation. Extensive experiments on various datasets demonstrate the effectiveness of **ISRec** compared with other state-of-the-art baselines and case studies show that we can identify dynamic user intents accurately.

ACKNOWLEDGMENT

This work was supported in part by National Natural Science Foundation of China Major Project (No. U1611461). All opinions, findings and conclusions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.
[2] R. He, W.-C. Kang, and J. McAuley, "Translation-based recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 161–169.
[3] R. He, C. Fang, Z. Wang, and J. McAuley, "Vista: a visually, socially, and temporally-aware model for artistic recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 309–316.
[4] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018.
[5] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1441–1450.
[6] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 565–573.
[7] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, "Context-aware sequential recommendation," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1053–1058.
[8] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2015, pp. 403–412.
[9] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 191–200.
[10] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 495–503.
[11] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.
[12] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.
[13] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1235–1244.
[14] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 689–698.
[15] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 305–314.
[16] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.
[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.
[18] Y. Cai, H.-f. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-based collaborative filtering recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 766–779, 2013.
[19] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[20] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 301–304.
[21] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.
[22] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 843–852.
[23] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 729–732.
[24] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.
[25] N. Zhu, J. Cao, Y. Liu, Y. Yang, H. Ying, and H. Xiong, "Sequential modeling of hierarchical user intention and preference for next-item recommendation," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 807–815.
[26] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, and X. Li, "Air: Attentional intention-aware recommender systems," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 304–315.
[27] S. Wang, L. Hu, Y. Wang, Q. Z. Sheng, M. Orgun, and L. Cao, "Intention2basket: A neural intention-driven approach for dynamic next-basket planning." IJCAI, 2020.
[28] 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*, 2020, pp. 1001–1010.
[29] M. M. Tanjim, C. Su, E. Benjamin, D. Hu, L. Hong, and J. McAuley, "Attentive sequential models of latent intent for next item recommendation," in *Proceedings of The Web Conference 2020*, 2020, pp. 2528–2534.
[30] T. Kipf, E. van der Pol, and M. Welling, "Contrastive learning of structured world models," *arXiv preprint arXiv:1911.12247*, 2019.
[31] T. Wang, R. Liao, J. Ba, and S. Fidler, "Nervenet: Learning structured policy with graph neural networks," 2018.
[32] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *arXiv preprint arXiv:1802.04687*, 2018.
[33] J. Kossen, K. Stelzner, M. Hussing, C. Voelcker, and K. Kersting, "Structured object-aware physics prediction for video modeling and planning," *arXiv preprint arXiv:1910.02425*, 2019.
[34] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 417–426.
[35] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," *arXiv preprint arXiv:1905.07854*, 2019.
[36] C. Zhang, K. Wang, H. Yu, J. Sun, and E.-P. Lim, "Latent factor transition for dynamic collaborative filtering," in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 452–460.

[37] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*, 2017, pp. 1263–1272.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[39] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 43–52.

[40] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, 2014, pp. 261–270.

[41] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, 2015.

[42] R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," in *The People's Web Meets NLP*. Springer, 2013.

[43] 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*, 2017, pp. 173–182.

[44] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[45] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[46] S. Rendle, "Evaluation metrics for item recommendation under sampling," *arXiv preprint arXiv:1912.02263*, 2019.

PLACE PHOTO HERE

Haoyang Li received the B.E. from the Department of Computer Science and Technology, Tsinghua University in 2018. He is a Ph.D. candidate in the Department of Computer Science and Technology of Tsinghua University. His main research interests focus on machine learning on graph-structured data, which has broad applications, ranging from social network analysis to recommender systems. He has published several papers in prestigious conferences, e.g., KDD, ICDM, etc.

PLACE PHOTO HERE

Xin Wang is currently an Assistant Professor at the Department of Computer Science and Technology, Tsinghua University. He got both of his Ph.D. and B.E degrees in Computer Science and Technology from Zhejiang University, China. He also holds a Ph.D. degree in Computing Science from Simon Fraser University, Canada. His research interests include cross-modal multimedia intelligence and inferable recommendation in social media. He has published several high-quality research papers in top conferences including ICML, MM, KDD, WWW, SIGIR etc. He is the recipient of 2017 China Postdoctoral innovative talents supporting program.

PLACE PHOTO HERE

Ziwei Zhang received his B.S. from the Department of Physics, Tsinghua University in 2016. He is currently pursuing a Ph.D. Degree in the Department of Computer Science and Technology at Tsinghua University. His research interests focus on network embedding and machine learning on graph data, especially in developing scalable algorithms for large-scale networks. He has published several papers in prestigious conferences and journals, including KDD, AAAI, IJCAI, and TKDE.

PLACE PHOTO HERE

Jianxin Ma received his B.E. from the Department of Computer Science and Technology, Tsinghua University in 2017. He is currently pursuing a master Degree in the Department of Computer Science and Technology at Tsinghua University. His research interests are mainly in machine learning, in particular representation learning, on relational data such as graphs. In particular, he strives to unify probabilistic generative modeling and deep representation learning, so as to accurately reveal the highly complex generating processes behind graph data and subsequently improve the robustness and interpretability of machine learning algorithms for graphs. He has published several papers in prestigious conferences, including KDD, AAAI, ICML, and NeurIPS.

PLACE PHOTO HERE

Peng Cui received the Ph.D. degree from Tsinghua University in 2010. He is currently an associate professor with tenure at Tsinghua University. His research interests include network representation learning, human behavioral modeling, and social-sensed multimedia computing. He has published more than 100 papers in prestigious conferences and journals in data mining and multimedia. His recent research efforts have received the SIGKDD 2016 Best Paper Finalist, the ICDM 2015 Best Student Paper Award, the SIGKDD 2014 Best Paper Finalist, the IEEE ICME 2014 Best Paper Award, the ACM MM12 Grand Challenge Multimodal Award, and the MMM13 Best Paper Award. He is an associate editor of IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Big Data, the ACM Transactions on Multimedia Computing, Communications, and Applications, the Elsevier Journal on Neurocomputing, etc. He was the recipient of the ACM China Rising Star Award in 2015.

PLACE PHOTO HERE

Wenwu Zhu is currently a Professor and Deputy Head of the Computer Science Department of Tsinghua University and Vice Dean of National Research Center on Information Science and Technology. Prior to his current post, he was a Senior Researcher and Research Manager at Microsoft Research Asia. He was the Chief Scientist and Director at Intel Research China from 2004 to 2008. He worked at Bell Labs New Jersey as a Member of Technical Staff during 1996-1999. He received his Ph.D. degree from New York University in 1996.

He served as the Editor-in-Chief for the IEEE Transactions on Multimedia (T-MM) from January 1, 2017, to December 31, 2019. He has been serving as Vice EiC for IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) and the chair of the steering committee for IEEE T-MM since January 1, 2020. His current research interests are in the areas of multimedia computing and networking, and big data. He has published over 400 papers in the referred journals and received nine Best Paper Awards including IEEE TCSVT in 2001 and 2019, and ACM Multimedia 2012. He is an IEEE Fellow, AAAS Fellow, SPIE Fellow and a member of the European Academy of Sciences (Academia Europaea).