Modeling Instant User Intent and Content-Level Transition for Sequential Fashion Recommendation

Yujuan Ding, Yunshan Ma, Wai Keung Wong, and Tat-Seng Chua

***Abstract*—Fashion recommendation, aiming to explore specific user preference in fashion, has become an important research topic for its practical significance to the fashion business sector. However, little work has been done on an important sub-task called sequential fashion recommendation, which aims to capture additional short-term fashion interest of users by modeling the item-to-item transitions. In this paper, we propose a novel Attentional Content-level Translation-based Recommender (ACTR) framework, which simultaneously models the instant user intent of each transition and the intent-specific transition probability. Specifically, we define instant intent with the relationships between adjacent items that the users interacted, which are the three fundamental domain-specific relationships of: *match*, *substitute* and *others*. To further exploit the characteristics of fashion domain and alleviate the item transition sparsity problem, we augment the item-level transition modeling with multiple sub-transitions using various content-level attributes. An attention mechanism is further devised to effectively aggregate multiple content-level transitions. To the best of our knowledge, this is the first work that specifies the implicit user actions in online fashion shopping with explicit instant intent, which enhances the connectivity of fashion items and boosts the recommendation performance. Extensive experiments on two real-world fashion E-commerce datasets demonstrate the effectiveness of the proposed method in sequential fashion recommendation.**

***Index Terms*—Fashion recommendation, Instant intent modeling, Translation method.**

I. Introduction

FASHION is one of the maturest E-commerce segments, and its global size is estimated at US$525.1 billion in 2019. The market is expected to grow further at 9.5% per year and reach a total market size of US$953.0 billion by the end of 2024.[1] Recommender systems can help fashion retailers optimise the conversion, average order value (AOV) and repeat purchase rate in E-commerce by understanding customers' styling preference and suggesting the right products to them. Driven by the great industrial demand and application potential, fashion recommendation has attracted increasing attention in data mining, information retrieval, and multimedia communities in recent years [1]–[5].

As an important problem in fashion recommendation, sequential fashion recommendation, which aims to capture both user-item interactions and item-item transitions, has not been well explored. Most personalized fashion recommendation works in the literature employ non-sequential approaches [1], such as collaborative filtering (CF)-based approaches [6], [7]. However, studies on sequential recommendation have shown that capturing the transition relationships within pairs of adjacent items (*i.e.*, item-to-item transition) in sequences is crucial in predicting the user's next action [8], [9]. In fashion domain, the item-to-item transition also plays a significant role in predicting user's next interacted items since his/her temporary fashion preference (*i.e.*, short-term fashion preference) swiftly changes and greatly affects the next action. However, modeling item-to-item transitions in fashion recommendation has not drawn sufficient attention in previous research [1], [10], [11]. Although various algorithms have been proposed in sequential recommendation [12], [13], directly applying those algorithms, which were originally designed for general domain or other specific domains, to address the problem in fashion domain can hardly achieve optimal performance due to domain gaps. In this paper, to develop advanced method specifically for fashion recommendation, we focus on two main problems in fashion domain: user's instant intent modeling and item's content-level attributes incorporation.

Instant intent widely exists behind each of user's behavior and affects his/her decision in the process of online fashion shopping. For example, after viewing a *black silk top*, a user might be interested in a *white silk skirt* in the following browsing since she would be thinking about the previous item and trying to find something to *match* it. In this case, when the user clicks the *white silk skirt*, her instant intent is *match*. Similarly, if the user has an instant intent of *substitute*, she will find another product with the same category, such as *blue silk top*. Based on such prevalent phenomena, considering users' instant intent in fashion recommendation becomes natural and rational. We are able to facilitate the recommendation from two aspects. First,

Manuscript received December 2, 2020; revised April 16, 2021; accepted May 13, 2021. Date of publication June 10, 2021; date of current version May 11, 2022. This work was in part supported by the Laboratory for Artificial Intelligence in Design (Project Code: RP3-1), Hong Kong, in part supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhu Liu. *(Corresponding author: Wai Keung Wong.)*

Yujuan Ding is with the Institute of Texitiles and Clothing, The Hong Kong Polytechnic University, kowloon Hong Kong, Hong Kong (e-mail: dingyujuan385@gmail.com).

Wai Keung Wong is with the Hong Kong Polytechnic University, kowloon, kowloon, Hong Kong, and also with Laboratory for Artificial Intelligence in Design, kowloon, Hong Kong (e-mail: calvin.wong@polyu.edu.hk).

Yunshan Ma and Tat-Seng Chua are with the School of Computing, National University of Singapore, Singapore 117416, Singapore (e-mail: yunshan.ma@u.nus.edu; dcscts@nus.edu.sg).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3088281.

Digital Object Identifier 10.1109/TMM.2021.3088281

[1] statista: eCommerce Report 2020-Fashion

different instant intents can pre-categorize the huge candidate item set, thus reducing the complexity of recommending the next item. For example, if we predict the intent to be *match* after clicking a *top*, we can lower the scores of or even exclude the items that cannot be matched with a *top*. Consequently, modeling the instant user intent before recommending the specific items can alleviate the sparsity problem introduced by the huge size of item set. Second, properly predicted intents can provide brief explanations for the recommendation results, thus enhancing the credibility of fashion recommender systems. In summary, modeling the instant user intent in fashion recommendation has great potential, while it has not been well explored in most previous fashion recommendation studies.

In addition to the user intent, the content-level attributes of fashion items also play a pivotal role in determining and explaining the user's interaction with items. Compared with general domains, where users' preference towards an item is mostly decided by the overall utility or functional properties of the item, in fashion domain, users' preference on the items is nuanced and more related to attributes. For example, when a user picks a *black silk top*, she might be interested in the *color*, *material*, *pattern*, *tailor* particularly, or any other design details. Therefore, incorporating such content-level attributes would help capture more specific user preference and enhance the modeling of sequential behaviours, which should be extensively investigated. Moreover, due to the extremely large number of fashion items, the sparsity of user-item interactions and item-item transitions becomes one of the most challenging issues in fashion sequential recommendation. With rich fashion attributes, additional dense semantic connectivities between items can be explored on top of the sparse CF signals, thereby further alleviating the severe sparsity problem to some extent. In short, content-level attributes of fashion items are promising to boost the recommendation performance if they are effectively incorporated, while they are undervalued and not properly modeled in previous fashion recommendation works.

In this work, to fill the above gaps in sequential fashion recommendation, we propose a framework named **A**ttentional **C**ontent-level **T**ranslation-based **F**ashion **R**ecommender (ACTR). We first define users' instant intent as the functional relationships between two adjacent items in the sequence of users' historical interactions, which are *match*, *substitute* and *others*. Specifically, the relationship between items is defined based on their categories. If two items a user successively interacts have exactly the same high-level category, the instant user intent in picking the second item is defined as *substitute*. If the categories of two adjacent items are complementary with each other, the instant intent is defined as *match*. If the instant intent of a user on two adjacent items does not belong to neither *match* nor *substitute*, it will be classified as *others*. After defining the instant user intent, we adopt a translation-based framework to simultaneously model the intent and the intent-specific item transition probability [9], [14]. As illustrated in Fig 1, the third-order interaction between the user u, the previous item i, and the next item j is not only simply modeled with a personalized translation operation (shown in (a)), but also specified with the incorporated instant intents (shown in (b)). To take into account

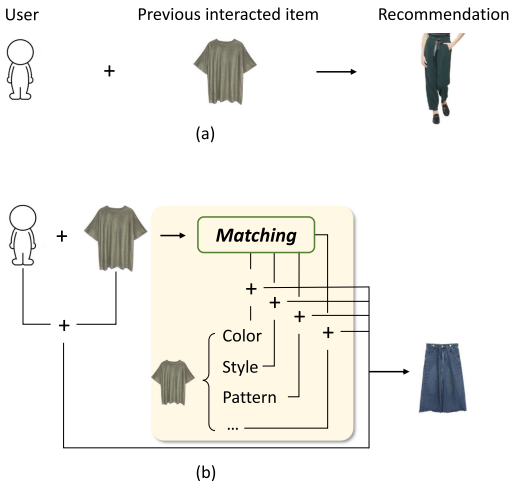

Fig. 1. Instead of solely modeling personalized item-item transition (showing in (a)), this study proposes to simultaneously model instant user intent (*match*, *substitute* or *others*) and the intent-specific recommendations. In addition, we augment the item-level transition model with multiple sub-transitions with respect to various content-level attributes, which further improves the recommendation performance.

the content-level attributes, we augment the translation model with multiple sub-transitions with respect to various content-level attributes. To effectively aggregate different sub-transitions regarding different fashion attributes, we adopt a user-aware attention mechanism over the multiple sub-transitions.

The contributions of this work are summarized as follows:

1) We propose to explicitly model instant user intent during online fashion shopping to enhance the sequential fashion recommender system in terms of both accuracy and interpretability.

2) We augment the typical item-to-item transition model with multiple sub-transitions using various content-level attributes. An attention mechanism is further devised to effectively aggregate multiple content-level transitions.

3) Extensive experiments on two real-world E-commerce fashion datasets (iFashion and Amazon) demonstrate the effectiveness of the proposed method in making preferable personalized fashion recommendation.

II. Related Work

In this section, we review two strands of research related to our work: fashion recommendation and sequential recommendation.

A. Fashion Recommendation

Fashion recommendation can be roughly grouped into two categories: recommendation based on a given user and recommendation based on a given item. The former task is also known as personalized fashion recommendation, which aims to explore user preference from the historical user-item interactions. Current studies on personalized fashion recommendation focus on enhancing models with visual features or leveraging aesthetic features [1], [4], [15]. Recommendation based on items, also

known as mix-and-match, usually works on compatibility modeling between items [2], [16]–[20]. For example, Han *et al.* proposed to use LSTM to model the complementary relationships among fashion items to make item-item recommendations [21].

In the task of personalized fashion recommendation, the model only focuses on the interaction between user and item, while the interaction between items is often ignored. In contrast, mix-and-match recommendation tasks only model the compatibility between items. Although some tasks such as personalized outfit recommendation [3], [10], [22], [23] explore both user-item and item-item relationships in the process of modeling, they model the compatibility relationships between items rather than the transition relationships resulted from user behaviors. Sequential recommendation (will be reviewed in detail in the following sub-section), which explores both the user-item interaction and item-item transition for modeling more comprehensive user preference, has not attracted much research attention in fashion domain [24]. In this paper, we aim to tackle the problem of sequential fashion recommendation. Note that both [24] and this work work on the sequential fashion recommendation task. However, since the two works were conducted almost at the same time, we do not compare the two methods. There exist differences in terms of data processing and experimental setting, please find details referring to the corresponding description of the two papers.

B. Sequential Recommendation

Sequential recommendation, which considers both user preference and sequential dependencies along the user's interaction sequence [25], is an important task in recommender systems [12]. Markov model-based, translation-based and deep neural network-based methods are three mainstream solutions for sequential recommendation. The Markov-based methods assume that user's next action only depends on the most recent preceding actions [13]. Factorized Personalized Markov Chain (FPMC) [8] is a representative Markov-based method, which models the interaction sequence as a first-order Markov Chain whose transition matrix is jointly factorized with a standard two-dimensional user-item matrix factorization approach. Some other Markov-based methods include PRME [26], HRM [27], etc [28]. The main limitation of this type of methods is that most of them only consider the low-order interactions (i.e., first-order and second-order), but ignore the possible higher-order ones [25].

Inspired by translational metric embeddings [29], many translation-based sequential recommendation methods have been developed in recent years. TransRec [9] unified item-item transition and user-item interaction to predict the next item a user might be interested in. MoHR [14] proposed to leverage heterogeneous item relationships in the TransE-based translation framework and achieved encouraging recommendation performance. CKE [30] utilized another translation model TransR [31] and incorporated multimedia knowledge to further boost the performance. The advantage of translation-based models compared with others is that they can easily incorporate additional information of various modalities into the sequential interaction modeling. This is also the reason why we employ TransRec as our basic model in developing our solution in this work.

The past few decades have witnessed the tremendous success of deep learning in many application domains [12], including recommender system. Various powerful deep neural networks have been successfully employed to develop the advanced sequence-aware recommender systems, such as recurrent neural networks (RNNs) [32], [33], convolutional neural networks (CNNs) [34] and graph neural networks (GNNs) [35], [36]. Specifically, RNNs can model the dynamics of interactions and sequential patterns of user behaviors, as well as various multimedia side information along with the sequential signal. Typical solutions include GRU4REC [33], NARM [37], STAMP [38] and others [39]–[41]. CNNs usually treat multiple interaction sequences as 2D images and they effectively extract patterns from these sequences [42]. GNNs are newly emerged methods in recommender systems in recent years, which are especially effective in modeling the complicated connectivities among users and items, as well as the dependencies in sequential behaviors, therefore enhancing the representative learning [43], [44]. Even though deep neural network-based recommendation solutions have achieved state-of-the-art performance in many specific sequential recommendation tasks, such as session-based recommendation [43], they require longer sequence samples to train the model and are reported to be not better than other factorization- or translation-based methods in several tasks and datasets [14], [45].

In summary, sequential recommendation has been an active research topic in general domain, where various types of recommender systems have been developed with different merits and demerits. However, directly applying existing methods can hardly achieve optimal performance in fashion domain since important domain-specific characteristics in fashion are overlooked. In this paper, we aim to enhance the sequential fashion recommendation model by considering user's instant intent and incorporating item's content-level attributes.

III. Problem Formulation

In this section, we formally formulate the problem as intent-aware sequential fashion recommendation. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{N_U}\}$ denote the whole user set, $\mathcal{I} = \{i_1, i_2, \ldots, i_{N_I}\}$ denote the whole item set, and $\mathcal{R} = \{r_1, r_2, \ldots, r_{N_R}\}$ denote the whole intent set. N_U, N_I and N_R are the total number of users, items and intents respectively. The fashion attributes[2] of items are also leveraged in the proposed method for modeling the content-level item-item transition. Specifically, we define the whole attribute set as $\mathcal{F} = \{f_1, f_2, \ldots, f_{N_F}\}$, where N_F is the number of attributes.[3] For each user $u \in \mathcal{U}$, we have an interaction list composed of successive items that u has interacted with $\mathbf{s}^u = [s_1^u, s_2^u, \ldots, s_t^u, \ldots, s_{|\mathbf{s}^u|}^u]$, and $s_t^u \in \mathcal{I}$. For each interaction list $\mathbf{s}^u$, there is a corresponding intent list $\mathbf{e}^u =$

[2]In this paper we use the word **attribute** to denote all associated fashion information we leveraged, including category, narrow sense attribute, and style.

[3]For simplicity, we omit subscript in the following part and use i, u, r, f to denote arbitrary item, use, intent and attribute.

$[e_1^u, e_2^u, \ldots, e_t^u, \ldots, e_{|\mathbf{e}^u|}^u]$, $e_t^u \in \mathcal{R}$, each instant intent e_t^u corresponds to a specific action s_t^u performed on an item.

For each attribute f, the value set of it is defined as $\mathcal{V}_f = \{v_1, v_2, \ldots, v_{N_f}\}$, where N_f is the number of values for attribute f. For example, *color* is an attribute, *red*, *black* and all other considered colors form the value set of the attribute *color*. Thus the whole value set of all attributes is $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{N_F}\}$.

In this work, we aim to learn a model, given a user, to simultaneously predict his next intent and the next item to be interacted based on the historical record. The input of the model includes the user u, the previous interacted item $i = s_t^u, t \in [0, |\mathbf{s}^u| - 1]$, as well as its attribute list $\mathbf{v}_i = [v_i^1, v_i^2, \ldots, v_i^{N_F}], v_i^f \in \mathcal{V}_f$. The outputs include two parts: 1) the probability of each intent $r \in \mathcal{R}$ to be the next intent $p(r|u, s_t^u)$ of user u at that specific time point (after interacted with s_t^u); and 2) the probability of each item $j \in \mathcal{I}$ to be the next interaction of the user $p(j|u, s_t^u)$.

IV. Approach

A novel **A**ttentional **C**ontent-level **T**ranslation-based Fashion **R**ecommender (ACTR) model is proposed in this paper to address the sequential fashion recommendation problem. The proposed ACTR model incorporates instant intent of users by predicting the instant intent and the intent-specific next interaction simultaneously. This model consists of three important parts: user instant intent prediction, content-level item transition modeling, and recommendation based on a mixture of transitions.

A. User's Instant Intent Modeling and Prediction

The user's instant intent we consider in this paper is the "reason" that the users may follow at a particular time point for a particular action. In the fashion context, such instant intent can be defined by the relationship of the adjacent items in an interaction sequence. For example, when a user interacted with two items successively which are both *boots*, his/her intent for the second action could be **substitute:***finding items with same function*, which is also the relationship of the two *boots*. Therefore, we explicitly define three types of user's instant intents: *match*, *substitute* and *others*. We assume that the user's intent in the process of fashion shopping is determined by the user's static tastes and his previous picks (in this paper we only consider the very last pick). Specifically, we apply a translational operation [29] to model the interaction between the user u, the previously interacted item i and the instant intent r:

$$R(r|u, i) = b_r - d(\boldsymbol{\theta}_u + \boldsymbol{\theta}_i, \boldsymbol{\theta}_r), \tag{1}$$

where $\boldsymbol{\theta}_u, \boldsymbol{\theta}_i, \boldsymbol{\theta}_r \in \mathbb{R}^c$ are the embeddings of u, i, r respectively and c is the embedding size. $d()$ denotes the distance measurement, which uses $L2$ distance in the specific implement of this work. b_r is the bias term related to the specific relationship r.

To obtain the probability of different instant intents under the condition of certain user u and his historical interaction i, a probability function P over all instant intents is defined as follows and applied:

$$P(r|u, i) = \frac{\exp(R(r|u, i))}{\sum_{r' \in \mathcal{R}} \exp(R(r'|u, i))}, \tag{2}$$

where $\mathcal{R}$ denotes the set of all intents. $P(r|u, i)$ can be interpreted as the possibility that the user u will adopt the intent r in his next action after interacting with item i.

B. Content-Level Intent-Aware Item Transition Modeling

Given an instant intent, i.e., an item-item relationships, we model the transition from item i to item j conditioned on intent r ($i \xrightarrow{r} j$) using a translational operation:

$$R(j|i, r) = b_j - d(\boldsymbol{\theta}_i + \boldsymbol{\theta}_r, \boldsymbol{\theta}_j). \tag{3}$$

However, such an approach is not sufficiently effective in our case as the item set is too large, which makes the item-item transitions very sparse. To enhance the modeling of item-item transition, it is necessary to leverage more content information of the fashion items rather than just using the item IDs. In fact, compared with other utility-oriented items, fashion items can be described in more details. Meanwhile, two items can be related with each other in various dimensions. For example, if A, B and C are interchangeable for being in the same category, B may be more similar with A than C if B and A share more similar details, such as the same color or same material. If the user cares more about the color or material in choosing similar clothes, he/she would prefer A rather than C after picking B.

Based on such consideration, we believe that the relationship r between two adjacent items i and j should be applicable not only at item-level, but also at the detailed content-level, *aka.*, the attribute-level. Given an item i, we have an attribute list $\mathbf{v}_i = [v_i^1, v_i^2, \ldots v_i^{N_F}]$ containing content-level information of the item. Instead of modeling the item-level transition (Eqn (3)), we model the intent-conditional item-to-item transition in specific attribute f with a translational operation:

$$R(j_f|i_f, r) = b_{j^f} - d(\boldsymbol{\theta}_i^f + \boldsymbol{\theta}_r, \boldsymbol{\theta}_j^f), \tag{4}$$

where $\boldsymbol{\theta}_i^f, \boldsymbol{\theta}_j^f \in \mathbb{R}^c$ denote the representations of attribute f for item i and j, *aka*, the embedding of v_i^f. For example, if f is *color*, the above operation can be interpreted as modeling the transition from item i to item j considering the color of the two items. Note that the item ID is treated as a special attribute and each specific item ID is one attribute value.

C. Attended Content-Level Transition Aggregation

There are several optional operations to aggregate the transition results of different attributes, such as average, sum or max pooling. Each operation represents a way to treat different attributes. For example, if we apply average or sum pooling, we assume that attribute values that belong to different attributes contribute equally to the item transition modeling. In comparison, max pooling only respects the dominant attribute and ignores the less influential attributes in the process of transition modeling. Instead of using the straightforward operations, this work

proposes a user-aware attention mechanism for the content-level transition aggregation.

The above attribute-specific item-to-item transition score $R(j_f|i_f, r)$ actually measures the transition probability of item i and j conditioned on the user intent r with regard to attribute f. Intuitively, the importance of different attributes should be different, and also, it would be affected by u who actually interacts with the two items. For example, when seeking for a matching item for the previously picked one, u may care more about the *style* but less about the detailed *patterns*. Based on such consideration, we propose to explore the user and previous item information to determine the importance coefficients of different attributes in the transition process. Specifically, inspired by graph attention network [46], the user-item-aware importance of attribute f is designed to be calculated as:

$$e_{f,ui} = \boldsymbol{a}^T[\boldsymbol{W}\boldsymbol{\theta}_u, \boldsymbol{W}\boldsymbol{\theta}_i^f], \tag{5}$$

where $\boldsymbol{W} \in \mathbb{R}^{c\times c}$ is the trainable weight matrix, $\boldsymbol{a} \in \mathbb{R}^{2c}$ is the trainable mapping vector. $[:,:]$ denotes the concatenation operation of two vectors. The final importance coefficient of f is calculated by applying a softmax function over all attributes:

$$\alpha_{f,ui} = \frac{\exp(e_{f,ui})}{\sum_{f'\in\mathcal{F}} \exp(e_{f,'ui})}. \tag{6}$$

Then the overall item-level transition under the intent r can be finally obtained by:

$$R(j|i, r) = \sum_{f\in\mathcal{F}} \alpha_{f,ui} R(j_f|i_f, r). \tag{7}$$

D. Sequential Recommendation

Given the probability of the intent that the user might have $\{P(r|u, i)\}_{r\in\mathcal{R}}$ (Section IV-A) and the intent-conditioned item-to-item transition scores $\{R(j|i, r)\}_{r\in\mathcal{R}}$ (Section IV-C), we can obtain the recommender results using weighted sum of the transition results of all intents. Specifically, the intent-aware fashion recommender score is calculated by:

$$R(j|u, i) = \sum_{r\in\mathcal{R}} P(r|u, i) \times R(j|i, r). \tag{8}$$

To better capture the dynamic transitions in user behavior sequences, we also model the implicit third-order interaction between u, i and j to capture the collaborative signals between users and items, which is specifically modeled by a personalized translation operation as:

$$R_0(j|u, i) = b_j - d(\boldsymbol{\theta}_u + \hat{\boldsymbol{\theta}}_i, \boldsymbol{\theta}_j). \tag{9}$$

We can understand the user embedding as the a special implicit intent to bridge two items, similar as the explicit intent we defined and leveraged in previous sub-section. Note that the item ID is the only attribute applied in modeling the user-item interaction, which means $\hat{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i^{id}$. Note that the embedding of $\hat{\boldsymbol{\theta}}_i$ is different from the $\boldsymbol{\theta}_i$ in Eqn (1).

The final recommendation result can be obtained by combining the implicit third-order transition with the explicit intent-aware transition modeling as follows:

$$R^*(j|u, i) = R_0(j|u, i) + \gamma R(j|u, i), \tag{10}$$

where γ is the hyper-parameter that balances the importance of two terms. Specifically, $R_0(j|u, i)$ emphasizes the long-term user preference and $R(j|u, i)$ focuses on the short-term sequential item-item transitions under various intents. Fig. 2 illustrates the inference process for modeling the interaction between the user u, previous item i and the following item j by considering explicit user intent and content-level transitions.

E. Model Training

As introduced in the previous section, the sequential fashion recommendation is formulated as a ranking problem in this study. Therefore, the final goal is to rank the ground truth next-item j higher than the irrelevant item j^-. The S-BPR [8] loss is used as the sequential recommender loss, which is defined as:

$$L_s = -\sum_{(u,i,j,j^-)\in\mathcal{D}_s} \ln(\sigma(R^*(j|u, i) - R^*(j^-|u, i))), \tag{11}$$

where $\mathcal{D}_s = \{(u, s_t^u, s_{t+1}^u, j^-)|u \in \mathcal{U} \cap t \in [|\mathbf{s}^u| - 1] \cap j^- \in \mathcal{I} - \mathbf{s}^u\}$. As the instant user intent is predicted in the process of user-item interaction modeling, the intent learning is also involved in the overall model learning. We apply a similar ranking loss to learn the intent as:

$$L_r = -\sum_{(u,i,r,r^-)\in\mathcal{D}_r} \ln(\sigma(P(r|u, i) - P(r^-|u, i))), \tag{12}$$

where $\mathcal{D}_r = \{(u, s_t^u, r, r^-)|u \in \mathcal{U}^{\tilde{}} \cap {}^{\tilde{}}t \in [|\mathbf{s}^u| - 1]^{\tilde{}} \cap {}^{\tilde{}}r = e_{t+1}^{u\,\tilde{}} \cap {}^{\tilde{}}r^- \in \mathcal{R} - r\}$.

Similarly, the item-relation-item transition loss is introduced in the training loss:

$$L_i = -\sum_{(i,r,j,j^-)\in\mathcal{D}_i} \ln(\sigma(R(j|i, r) - R(j^-|i, r))), \tag{13}$$

where $\mathcal{D}_i = \{(i, r, j, j^-)|i \in \mathcal{I}^{\tilde{}} \cap {}^{\tilde{}}j \in \mathcal{I}_{i,r}{}^{\tilde{}} \cap {}^{\tilde{}}j^- \in \mathcal{I} - \mathcal{I}_{i,r}\}$. $\mathcal{I}_{i,r}$ denotes the item set that consists of items having relationship r with item i. Finally, the problem eventually becomes a multi-task learning problem, with the overall loss as:

$$L = L_s + \alpha L_r + \beta L_i, \tag{14}$$

where α and β are hyper-parameters that balance the importance of different tasks.

F. Discussion

The ACTR model borrows the original idea from TransRec [9] which applies a single translation operation to effectively model the third-order interaction between u, i and j. It is closer to MoHR [14] model in terms of translation-based interaction modeling and the basic multi-task learning framework. Also, ACTR and MoHR both propose to build additional item-item connectivities by leveraging item relationships. But ACTR is different

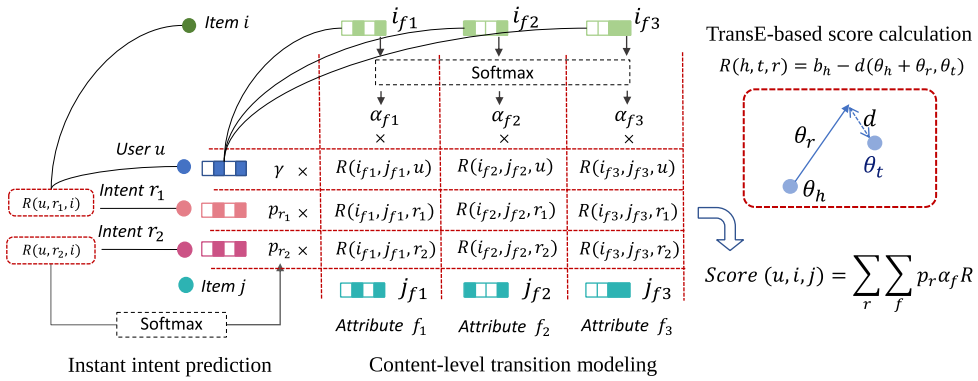

Fig. 2. Illustration of proposed attentional content-aware translation-based fashion recommender (ACTR). It demonstrates how to calculate the preference score for the next item given the user and his previous interacted item.

from MoHR mainly in three aspects. First, MoHR uses the same item embedding to predict the next relationship and model the user-item interaction. In other words, with the same user representation $\boldsymbol{\theta}_u$ and item representation $\boldsymbol{\theta}_i$, the model tries to predict the next item j and next relationship r in the same manner, which will falsely guide the representation learning of j and r. is unreasonable and limits the ability of the entire model. In ACTR, to avoid this problem, two different item representations are employed for different tasks respectively ($\boldsymbol{\theta}_i$ for predicting relationship in Eqn (1) and $\hat{\boldsymbol{\theta}}_i$ for predicting the next item in Eqn (9)). Second, the content-level item-to-item transition is modeled in ACTR to better take advantage of available information specifically in fashion domain and further alleviate the sparsity problem in the item-item transition. In contrast, all item transitions in MoHR are modeled based on ID solely. Third, a novel attention mechanism is introduced in the content-level transition aggregation, through which the importance of different attributes are determined by the user and previous item. The MoHR method is included as one of the baselines in this study; the experimental results show the different performance of the proposed ACTR and MoHR methods. All technical improvements in ACTR as we discussed will be evaluated in the ablation study.

Time complexity analysis The time complexity of the training procedure is $O(NB(N_F c^2 + (N_F N_R + 1)c))$, where N is the number of iterations and B is the batch size. N_F and N_R are the number of attributes and intents respectively. c is the size of embeddings. The time complexity of the closest baseline MoHR is $O((N_R + 1)c)$. Compared to MoHR, the proposed ACTR method increases the number of parameters, as well as the time complexity. However, the performance of the method greatly boosts under the condition of limited time complexity increase. The average improvement of ACTR over MoHR is more than 10% on all three metrics on both four experimental settings. The time complexity increase is mainly caused by the attention mechanism introduced in ACTR. Without applying the attention mechanism, the time complexity would be $O((N_F N_R + 1)c)$. As N_F is relatively small, the time complexity increase over MoHR is not quite significant, but such a method also clearly outperforms MoHR according to our experimental results in the ablation study. In summary, the proposed ACTR is not the most efficient method. However, we can see that considering the great improvement on its recommendation performance, such an increase in time complexity is acceptable. Moreover, with the acceleration using multi-core CPUs and GPUs, the method can be implemented more efficiently.

V. Experiments

In this section, we conduct experiments on real-world fashion E-commerce datasets to evaluate the proposed method for sequential fashion recommendation. In particular, we aim to answer the following research questions:

- **RQ1:** How does the ACTR method perform in the fashion recommendation tasks compared to other state-of-the-art recommendation methods?
- **RQ2:** Does each component devised in the ACTR model contribute to the recommendation performance improvement?
- **RQ3:** How specifically do the introduction of instant user intent and the content-level item transition modeling help in refining the fashion recommendation results?

A. Datasets

We evaluate our ACTR method on two large-scale fashion datasets, iFashion [47] and Amazon [48]. The two datasets are both collected from E-commerce platforms. Specifically, iFashion dataset is from Taobao[4] and originally proposed for outfit generation and recommendation. It is composed of two parts: one is the user clicking records in short sessions, in which each piece of record contains the user and item list he/she interacted with in chronological order. Another part of iFashion is the user-outfit interaction data, which contains the information of the user, the outfit he/she interacted with, and the items in the outfit. We employ the first part in this paper for the sequential fashion recommendation task, *aka*, the user clicking records of fashion items. As the original dataset is extremely large (with over three millions of users and over four millions of items), for this study, we generate a subset for iFashion-Sequential Recommendation (iFashion-SR) from iFashion.

[4][Online]. Available: http://www.taobao.com/

TABLE I
DATASET STATISTICS

Dataset	iFashion-SR		Amazon-Fashion	
Seq Length	Five	Eight	Five	Eight
#User	36,752	36,797	92,276	41,878
#Item	458,642	460,596	477,912	355,571
#Train sample	1,324,637	1,188,988	515,978	301,427
#Test sample	50,001	50,002	92,276	41,878
#Valid sample	50,001	50,002	92,276	41,878
User-item density	7.86e-5	7.02e-5	1.17e-5	2.02e-5
Item-item density	6.27e-6	5.60e-6	2.26e-6	2.38e-6

Amazon dataset is a large-scale review dataset which consists of purchase records, reviews for products on Amazon,[5] as well as the meta information and images of products. The user behavior sequence of Amazon dataset is generated by the user's purchased items in chronological order. We only use the Clothing-Shoes-Jewelry subset of the Amazon data (2014 version) in our experiments and name it as Amazon-Fashion.

For fair evaluation of all sequential recommendation methods, the dataset should be applicable to different sequential recommendation methods. Considering that most deep neural network-based sequential recommendation methods [42], [43] take a fixed length of historical interaction sequence as model input, we apply the sliding window strategy [42] on each long sequence to generate short sequences with fixed length. Specifically, two settings are designed with the window length equaling to **seven** and **ten** to make the evaluation more comprehensive. We use the last and the second to last short sequences in each long sequence for testing and validation respectively; and all the remaining short sequences are used for training. To make sure that each long sequence can generate at least one test, validation, and training sequence, we abandon sequences shorter than **seven** and **ten** respectively. For iFashion, we keep approximately 50 000 qualified user-item interaction sequences under both settings. Note that one user can have more than one interaction sequences, so that the sequence quantity is more than user quantity for iFashion (Referring to Table I). For Amazon, we keep all qualified sequences.

The specific data statistics of the two datasets are shown in Table I, and each of them has two settings. For iFashion, we do not filter out items based on their interaction frequency, while for Amazon-Fashion we remove items that have less than five interactions since the item interaction of this dataset is too sparse. As shown in Table I, the scale of item set for both datasets are clearly larger than that of the user set. The item-item density is very small, especially for Amazon, which shows that the item-item sparsity problem exists in real-life datasets in fashion recommendation.

After fixing two basic datasets, images of all involved items are downloaded and a commercial fashion tagging tool Visenze[6] is applied to extract three types of fashion attributes (category, narrow sense attribute and style) from images. A total of 225 different fashion attribute values (such as *dress*, *red*) are included, which belong to 24 attributes based on a certain fashion taxonomy.[7] Attributes include *category*, *style*, and narrow sense attribute such as *color*, *pattern*, *neckline style* and *dress shape*. Attribute values belonging to various attributes include *color:white*, *style:sexy*, *dress shape:A-line*, *etc*.

B. Extraction of Instant User Intent

In this part we introduce how we extract the instant user intent, i.e., the item relationships (*match*, *substitute* and *others*) in the datasets. As introduced previously, we define the item relationships only at function-level, which is based on the category information. The Visenze taxonomy, which all our item attributes are generated based on, provides a hierarchical three-level category information, such as [*Apparel→Upper body garment→Shirt*], [*Apparel→Full body garment→Dress*]. If two items have exactly the same three-level category, they are interchangeable with each other, therefore holding the *substitute* relationship. For example, two [*Apparel→Upper body garment: Shirt*]s are *substitute*, [*Apparel→Upper body garment→Shirt*] and [*Apparel→Upper body garment→T-Shirt*] are not *substitute*.

The definition of *match* relationship is more complicated as two items with different categories are not necessarily complementary with each other. We therefore define the *match* relationship as two categories that are often combined together in good-quality outfits. For example, [*Apparel→Upper body garment→T-shirts*] and [*Apparel→ Lower body garment→Jeans*] are matching, while [*Apparel→Lower body garment→Jeans*] and [*Apparel→Lower body garment→Skirts*] are not. We extract such matching knowledge with regard to categories from external fashion outfit datasets. Specifically, we adopt the iFashion outfit dataset [5] mentioned above which consists of outfits generated by fashion experts from Taobao. Similarly, we apply the Visenze tagging tool to tag the fashion items in each outfit to obtain the category information. We then calculate the frequency of each unique category pair existing in the outfit and extract the most frequent category pairs based on a proper criteria. The extracted frequent category pairs form the *match category pair set*. Given any two items, if the pair of their corresponding categories are in the *match category pair set*, the two items are considered to keep the *match* relationship.

Item pairs that are not *match* nor *substitute* are regarded as keeping the *others* relationship. For example, the relationship of above mentioned *Jeans* and *Skirts* is classified as *others*.

C. Experimental Settings

Baselines: We compare the proposed ACTR method with several competitive and relevant baselines:

[5][Online]. Available: http://www.amazon.com/

[6]visenze.com

[7]We use the ViSenze taxonomy.

- **MF** [6]: This is the basic Matrix Factorization model using BPR loss. MF is the classic non-sequential recommendation model which is easy to apply and very powerful in certain applications.
- **FMC** [8]: It focuses on modeling the sequential dynamics by factorizing the item-item transition matrix, which ignores the personalization in the sequence modeling.
- **FPMC** [8]: It models both personalized user-item interactions and the "global" item-item transitions by MF and FMC respectively, which therefore captures the personalized Markov behavior.
- **HRM** [27]: This method extends FPMC by using aggregation operations such as max pooling to model more complex interactions. Here, we specifically use the max pooling since its performance is more competitive than average pooling according to the experimental results.
- **PRME** [26]: It models the personalized Markov behavior by the sum of two Euclidean distances rather than the inner product used in FPMC.
- **FM** [49]: It models pairwise feature interactions as inner product of latent vectors between features to estimate the user behavior on items.
- **NFM** [50]: It enhances FM by modelling the higher-order and non-linear feature interactions.
- **Caser** [42]: It treats the embedding of historical interacted items as an "image" and proposes to use CNN to capture both personalized preference and sequential patterns.
- **GRU4REC** [33]: This is an RNN-based sequential recommendation method. It also processes fixed length of historical items and applies GRU to handle the sequential data. It does not model user information and only explores the sequential signal while ignoring personalized preference.
- **SR-GNN** [43]: This is a GNN-based sequential method that leverages the transition relationships between items in the sequence to enhance the recommendation performance.
- **TransRec** [9]: This is the basic translation-based recommendation approach which unifies user preferences and sequential dynamics in a single translation operation.
- **MoHR** [14]: MoHR uses various recommenders to capture long-term user preferences and short-term item transitions in a unified translational metric space.

In summary, all the baselines can be classified into four groups, namely matrix factorization-based methods, deep learning-based methods, translation-based methods and attribute-involved methods. MF, FMC, FPMC, HRM, PRME are matrix factorization-based methods, which are also the mainstream recommender systems. Caser, GUR4REC and SR-GNN are representative deep learning-based recommenders that employ CNN, RNN and GNN respectively. SR-GNN is the state-of-the-art methods proposed recently. TransRec and MoHR are both translation-based methods which are close to the proposed ACTR model. The other two, FM and NFM are representative methods which train the recommender system not only based on the interaction records but also the attributes.

Implementation Details: For fair comparison, the last item of each sequence is treated as the target item for recommendation in the proposed ACTR method and all baselines. For all translation-based and matrix factorization-based methods, including ACTR, only the last two items in each sequence are used in training the model, which are the previous and next items. For all deep learning-based methods, *i.e.*, Caser, GRU4REC and SR-GNN, the entire item sequences are used. Specifically, all items except the last one are used as the historical input while the last one is the item to recommend. Such implementation ensures that the quantity of training data and prediction targets are the same for all comparing methods[8]. For ACTR, the hyper-parameters α, β and γ are set to 0.1, 0.1 and 0.5 by fine-tuning the model and referring to MoHR [14]. The embedding size of all types of embeddings is set to 10 for iFashion and 5 for Amazon based on empirical testing. For the training process, the learning rate is set to 0.005 and the batch size is set to 5000. The maximum training epoch for ACTR is set to 4000. All baselines are carefully tuned with different training settings including the learning rate, batch size and weight decay. All settings for baselines refer to the reference papers for best performance.

Evaluation Metrics: For the main evaluation, we randomly sample 100 negative items for each test and validation sample [51], following the mainstream resaerch in recommendation [14], [52]. The negative items are not interacted by the corresponding user. However, there have been doubts in recent research that sampled metrics with small negative sets might not reflect the real trends of performance faithfully [53]. In our case, the item set is too large for both datasets, which makes the evaluation with full item list ranking [54] not applicable as it will result in large computational cost. Additionally, the metric values would be too small to effectively evaluate all methods. Nevertheless, to make more comprehensive evaluation on our method, we have a **second** evaluation setting with larger and multiple negative sets paired with each positive sample. Specifically, we randomly sample **five** sets of **1000** negative items for each test and validation sample. Three common top-K evaluation metrics are employed to quantitatively evaluate the effectiveness of methods: RECALL@K, NDCG@K and MRR@K. K is set to 10 by default. The results for all metrics are reported averagely.

D. Fashion Recommendation Performance (RQ1)

Tables II shows the overall recommendation performance of the proposed ACTR method and all the baselines on the iFashion-SR and Amazon-Fashion datasets, each of which has two settings of sequence length. The observations from the results are as follows:

1) Overall, the proposed ACTR method shows competitive performance on both datasets for both two settings. On iFashion-SR, the proposed ACTR achieves the best performance among all compared methods while on Amazon-Fashion, the ACTR

[8]Note that the data setting here differs with that in [24]. All compared methods have the same number of training, testing and validation samples in this paper. However, in [24], item pair-based methods, e.g., FMC, FPMC, have more training samples since the first several item pairs in each long sequence are kept and used. The data settings for the item sequence-based methods, e.g., Caser, are same in this work and [24]

TABLE II
RECOMMENDATION PERFORMANCE OF THE ACTR MODEL AND THE BASELINES

Dataset	iFashion-SR						Amazon-Fashion					
Seq Length	Five			Eight			Five			Eight		
	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall
MF	0.3602	0.4023	0.5377	0.3519	0.3934	0.5266	0.1684	0.1911	0.2655	0.1052	0.1250	0.1903
FMC	0.2934	0.3259	0.4313	0.2895	0.3215	0.4253	0.1254	0.1459	0.2130	0.0711	0.0868	0.1388
FPMC	0.3615	0.4037	0.5396	0.3540	0.3952	0.5277	0.1668	0.1904	0.2676	0.1039	0.1236	0.1886
PRME	0.3306	0.3661	0.4804	0.3210	0.3556	0.4672	0.0892	0.1158	0.2035	0.0622	0.0853	0.1621
HRM	0.3425	0.3842	0.5185	0.3369	0.3789	0.5131	0.1882	0.2124	0.2917	0.1231	0.1446	0.2156
FM	0.2627	0.3076	0.4537	0.2470	0.2854	0.4086	0.1440	0.1726	0.2668	0.1031	0.1274	0.2068
NFM	0.3050	0.3528	0.5086	0.2960	0.3447	0.5030	0.2077	0.2383	0.3377	0.1637	0.1876	0.2662
GRU4REC	0.3586	0.4021	0.5418	0.3574	0.4010	0.5413	0.2015	0.2260	0.3058	0.1363	0.1582	0.2300
Caser	0.2596	0.2931	0.4011	0.2554	0.2885	0.3952	0.1797	0.2041	0.2843	0.0730	0.0940	0.1639
SR-GNN	0.3862	0.4248	0.5485	0.3890	0.4271	0.5494	**0.2621**	**0.2833**	0.3518	**0.1867**	**0.2076**	0.2757
GCE-GNN	0.3721	0.4088	0.5268	0.3657	0.4031	0.5228	0.1980	0.2211	0.2963	0.1460	0.1683	0.2413
TransRec	0.2921	0.3137	0.3843	0.2898	0.3106	0.3791	0.0850	0.1061	0.1761	0.1017	0.1233	0.1948
MoHR	0.4177	0.4519	0.5629	0.4082	0.4415	0.5495	0.1895	0.2121	0.2863	0.1309	0.1525	0.2242
ACTR	**0.4844**	**0.5229**	**0.6468**	**0.4751**	**0.5122**	**0.6314**	0.2106	0.2491	**0.3745**	0.1550	0.1935	**0.3200**

method outperforms all compared methods except for the SR-GNN under the MRR and NDCG evaluation. Such experimental results demonstrate the effectiveness of the ACTR method in the fashion recommendation task.

2) Comparing the results of MF, FMC and FPMC, we observe that the performance of FMC is much worse than that of MF. Moreover, although FPMC achieves better performance than the other two overall, its performance is very close to that of MF. Recall that FMC only models the item-item transitions while MF only models the user-item interactions, such experimental results show that item-item transitions are more difficult to model in our cases, which is probably due to the sparsity problem pointed out in the previous analysis. That also explains why FPMC only outperforms MF by a slight margin although it models both item-item and user-item interactions and is supposed to be much better. These results also justify the motivation of ACTR which is to enhance the item-item transition modeling by leveraging the transition at the content level.

3) Comparing the result of TransRec with MoHR, we can see that by incorporating the instant user intent into modeling the item-to-item transition, the recommendation performance of the translation-based method can be vastly improved. The TransRec shows the worst performance in our experiments, which indicates that the simple single component translation model might not be effective enough to model the complex interaction patterns in our fashion recommendation tasks. This also explains why the introduction of the instant intent is necessary, as well as the content-level transitions.

4) Overall, the Amazon-Fashion dataset is more challenging than the iFashion-SR according to the performance of all methods on two datasets. Such results are reasonable as the Amazon-Fashion has much fewer training samples but more users and items than iFashion-SR according to Table I. By comparing different types of basic models, we find that matrix factorization-based methods achieve more competitive performance on iFashion-SR than the other three groups of methods. On Amazon-Fashion, the deep learning-based methods, especially the GRU4REC and SR-GNN, perform better than the other three types of methods in general. Since the item-item transitions are noisy due to the long time span of Amazon, it is more challenging to model such transitions. The deep learning-based methods perform better because they captures the long-term preference.

5) The experimental results between the two settings of sequence length are not markedly different for iFashion-SR since the statistics of the two subsets are similar, i.e., both of them are generated from the original large-scale iFashion dataset. But we can still notice that most methods perform slightly better in the first experimental setting (sequence length=7), which might result from more training samples. The difference in performance between the two settings is more obvious on the Amazon-Fashion dataset as the statistics of the two settings are very different.

6) The performance of ACTR method is not as stable and preferable on Amazon-Fashion as that on iFashion-SR. Even though it outperforms all comparing methods in terms of the Recall rate, it shows worse MRR and NDCG performance compared to SR-GNN. The reason may be multi-fold. First, the explicit instant user intent does not widely exist in this dataset, which makes it hard to explore and take advantage. More specifically, the category-level relationships between items in Amazon-Fashion might be not consistent with the collaborative signals so that modeling such relationships cannot actually facilitate the user-item interactions or item-item transitions modeling much. Such a reason explains why MoHR also shows bad performance on this dataset. When thinking of the problem more deeper, we find such results to be reasonable since the Amazon-Fashion dataset is composed of purchase records, in which the *instant*

TABLE III
PERFORMANCE UNDER LARGE NEGATIVE SET OF EVALUATION ON IFASHION-SR

Seq Length	Five			Eight		
	MRR	NDCG	Recall	MRR	NDCG	Recall
FPMC	0.3527	0.2326	0.2610	0.3453	0.2269	0.2553
GRU4REC	0.3279	0.2163	0.2427	0.3216	0.2120	0.2378
MoHR	0.3824	0.2988	0.3186	0.3707	0.2934	0.3117
ACTR	0.4494	0.3400	0.3660	0.4356	0.3292	0.3545

intent of users might be weak. Instead, the deep learning-based methods show preferable performance, especially the SR-GNN, for being able to model not just the previous one behavior but the whole session. Such results demonstrate that modeling the implicit dependency in a period of time is important in the sequential recommendation problems when the interval of behaviors is comparably large.

Performance comparison with the large negative set of evaluation In table III, we report the performance of our method and the three most competitive baselines under the second evaluation scenario (referring section V-C) on iFashion-SR dataset. From the results, we can see our ACTR still significantly outperforms baselines under the evaluation with more negatives, further demonstrating its effectiveness.

E. Ablation Study (RQ2)

We further conduct ablation studies on the two datasets to validate the effectiveness of several novel components devised in the ACTR model. In particular, three technical components are investigated. First, compared with the MoHR method, how does it help by applying two different sets of item embeddings for predicting the instant intent and modeling the user-item interactions respectively. Second, whether the modeling of content-level item-item transition helps improve the recommendation performance as expected. Lastly, whether the attention mechanism designed to determine different importance of various attributes to different users further improve the recommendation performance. The results of ablation study on the two datasets are reported in Tables IV. **Vanilla** denotes the basic ACTR model that only uses one set of item embeddings, no content-level transition modeling, and no attention mechanism applied, which is same as the MoHR model. **V+T** denotes the Vanilla model but with two sets of item embeddings, **V+C** denotes the Vanilla model but with content-level transition modeling and **V+T+C** denotes the model that has both components (refer to section IV-F for a detailed description). The final ACTR model is equipped with attention mechanism for the content-level transition aggregation based on the **V+T+C** model.

The results show that all novel technical components in ACTR work on iFashion-SR and improve the recommendation performance on both experimental settings. On Amazon-Fashion, the setting of two sets of item embedding does not bring much improvement in the experiments where the sequence length is set to ten. In the process of experiment, we observe that the ACTR method is easier to overfit on Amazon-Fashion than on iFashion-SR, which we believe is because the Amazon-Fashion dataset has much fewer training samples. Based on such experimental observation, we chose the smaller embedding size for ACTR on Amazon-Fashion and achieve the desired performance improvement eventually. Among all four experimental dataset settings, Amazon-Fashion with sequence length of ten has the least training samples, which makes it the most challenging setting and the Vanilla model is easy to overfit for this setting. Under this circumstance, adding extra set of item embeddings would make the model more complicated and harder to train. This explains why the strategy of two sets of item embeddings failed on this setting. Overall, the content-level transition modeling and the user-aware attention mechanism greatly boost the recommendation performance, reflecting that enhancing the item-item transition is important in sequential recommendation. In the fashion domain, such intent-specific item-item transition modeling can be improved by specifying and leveraging detailed information from different fashion aspects.

We further conduct parameter analysis on three hyper-parameters α, β and γ to discuss their influence on the overall performance of ACTR. Referring to [14], we test each parameter when the other two are fixed. The experimental results are illustrated in Fig. 3 (a). As we can see, our method is not quite sensitive to hyper-parameters since the performance is stable when the parameters are set properly in a certain range.

F. Qualitative Analysis (RQ3)

In this section, we conduct more qualitative analysis based on the experimental results on iFashion-SR (sequence length=7) to discuss the proposed method in detail and show the merits of the method more explicitly. First, in Fig. 3 (b), we show the relationship between the normalized probabilistic preference score based of the ground-truth intent with the ranking results of testing samples (the NDCG score). Specifically, the normalized probabilistic preference score of the ground-truth intent denotes the contribution of the right intent in the overall preference scores, which is calculated by

$$S_{r_{gt}} = \frac{P(r_{gt}|u,i) \times R(j|i,r_{gt})}{\sum_{r \in \mathcal{R}} P(r|u,i) \times R(j|i,r)}. \tag{15}$$

From the figure we observe that the performance of the recommendation results improves consistently when the contribution of the ground-truth intent in the preference scores increases. This indicates that the more the contribution the correct intent makes, the better the overall recommendation performance is. When $S_{r_{gt}}$ is close to 1, which means other intents barely affect the final preference score, the NDCG results are much higher than those test samples with lower $S_{r_{gt}}$. Such results show that exploring the instant user intent facilitates the sequential recommendation tasks as the prediction results based on the correct intent is positively correlated with the final recommendation results.

Next, in Fig. 4, we illustrate two examples with predicted intent probability, as well as the final top 10 ranking lists of the TransRec and our ACTR method. Since our ACTR employs the same basic translation-based model as TransRec but leverages

TABLE IV
RESULTS OF ABLATION EXPERIMENTS

Dataset	iFashion-SR						Amazon-Fashion					
Seq Length	Five			Eight			Five			Eight		
	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall
Vanilla	0.4177	0.4519	0.5629	0.4082	0.4415	0.5495	0.1895	0.2121	0.2863	0.1309	0.1525	0.2242
V + T	0.4272	0.4600	0.5661	0.4155	0.4474	0.550	0.1927	0.2154	0.2902	0.1303	0.1519	0.2235
V + C	0.4399	0.4761	0.5935	0.4258	0.4609	0.5752	0.1886	0.2275	0.3546	0.1292	0.1628	0.2732
V + T + C	0.4449	0.4791	0.5897	0.4311	0.4652	0.5758	0.1974	0.2345	0.3553	0.1128	0.1474	0.2621
ACTR	**0.4844**	**0.5229**	**0.6468**	**0.4751**	**0.5122**	**0.6314**	**0.2106**	**0.2491**	**0.3745**	**0.1550**	**0.1935**	**0.3200**

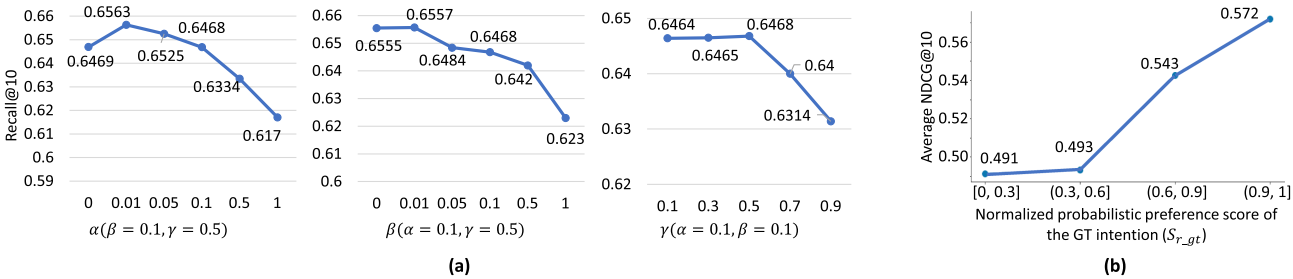

Fig. 3. (a) Effect of hyper-parameters α, β and γ. Experiments are conducted on iFashion-SR (seq_len = 5). (b) Intent-specific recommendation based on the predicted instant user intent.

Fig. 4. Visualization of the predicted instant intent and the Top 10 ranking lists of the TransRec and the proposed ACTR method. The ground truth intent of each item is shown above the item image.

the instant user intent modeling and the content-level transitions additionally, by comparing with the TransRec, we try to illustrate the difference qualitatively made by the the two main contributions of ACTR. In the first case, given the user and the previous item, we can see that the intent of the user in picking the next item is predicted more likely to be *match*, less likely to be *others*, but not possible to be *substitute*. In the second case, the most possible user intent for the next item is predicted to be *match* too. In the ranking list, besides the top 10 items, we also show the corresponding intent of each item (the relationship between this item and the item the user previously chose, which is the anchor shown in the very left) above the item image. From

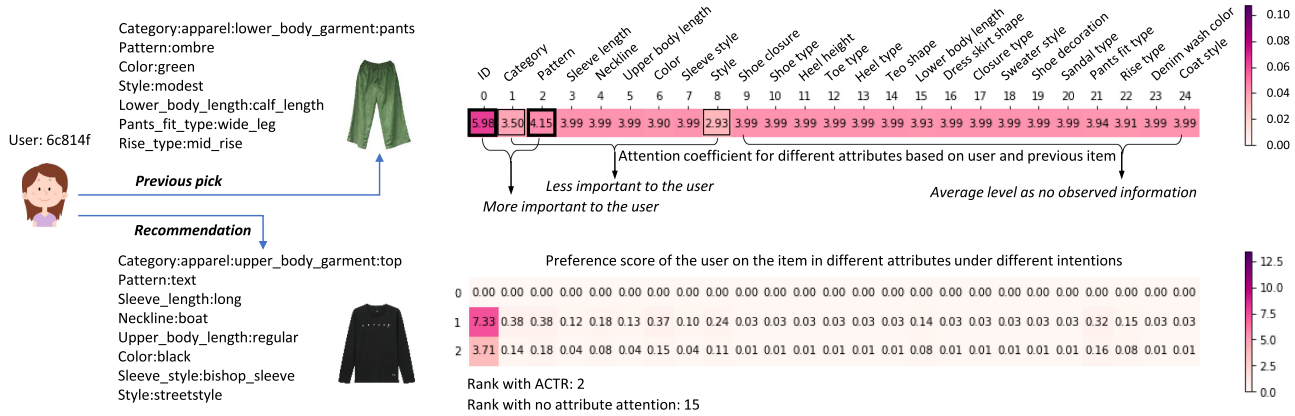

Fig. 5. Illustration of the attribute attention coefficients based on the user and previous item (top-right) and the disentangled intent-aware preference scores (bottom-right).

(a) Recommendation for matching **(b)** Recommendation for substitute

Fig. 6. Intent-specific recommendation based on the predicted instant user intent.

the ranking results, we can see that first, our method is able to rank the ground-truth next-item higher than TransRec in both cases. Such improvement in ranking results between ACTR and TransRec demonstrates the effectiveness of the technical improvements of our method. Moreover, we can also observe that the ACTR ranks more items corresponding to the correct intents in the top 10 list than the TransRec.

In Fig. 5, we show a case with the visualization of the preference score calculation of the candidate item given the user and the previous item. On the left side, we illustrate the user, the previous item he/she interacted, and the next item which might/might not be the user's next interaction, as well as the attribute values of the items. On the top-right, we show the attribute-level attention scores which depict the importance of each attribute based on the user and the previous item. First of all, we can see that the user cares about different attributes differently. Specifically, the *ID*, *Pattern* are learned to be more important to the user while other attributes such as *Category* and *Style* are less important. For other attributes that are not observed in the user's last interacted item, the importance weights of them (attention coefficient) are at the same average level, which is determined by the user him/herself only. The calculation of attention coefficient in this part is introduced in Eqn (5) and Eqn (6).

On the bottom-right, we further illustrate detailed preference scores of the user on the candidate items corresponding to various instant intents of the user (explicit relationship between the two items). The horizontal axis denotes attributes while the vertical axis denotes the three explicit intents. Specifically, each square in the heat map represents a score corresponding to specific intent and attribute, which shows the probability of the item being picked considering one specific relationship between it and the previous item with regard to one specific attribute. From the presented case, we can see that several scores under the *match* intent are comparatively higher, including *ID*, *Category*, *Pattern* and *Color* while the scores of *substitute* are lower. Such results tell us that the candidate item is quite well matched with the previous item, especially in terms of *category*, *pattern* and *color*. All scores under the intent of *others* are zero, which shows the user's intent at this time point is predicted to be either *match* or *substitute*, but unlikely to be *others*.

At the bottom of the preference score map, we also show the rank of the ground-truth item (the present candidate) in the ranking list, which is 2 by the ACTR model but 15 by the ACTR model without attention coefficients (the **V+T+C** model discussed in Sec. V-E). This case demonstrates that applying the user-aware attention mechanism in aggregating the content-level transitions, which learns the different importance of various attributes to users, can clearly improve the overall performance of fashion recommendation and rank the right item higher in the final recommendation list.

By effectively predicting the instant intent, we can manipulate the ranking results manually to only recommend the items that are in accordance with the predicted intent. For example in Fig. 6, the user in the first case is predicted by the ACTR model that he wants the items to *match* with his previous pick. Therefore, the final recommendation list contains matching proposals only. Likewise, in the second case, the user's intent for the next item is

predicted to be *substitute*. As a result, a recommendation list full of similar items is presented to him/her. Such recommendation is more specific and customized, which is able to help users engage in the process of online shopping and improve the conversion.

VI. Conclusion

This paper worked on the sequential fashion recommendation task, aiming to simultaneously model the instant user intent and the intent-specific item-to-item transition. We explicitly defined the user intent in fashion domain as the relationship of adjacent items he/she interacted, which is *match*, *substitute* or *others*. Experimental results on two real-world fashion E-commence datasets demonstrated the effectiveness of intent modeling in improving the fashion recommendation in terms of accuracy and explanability. To take better advantage of the domain characteristics and further specify the item-to-item transition, we proposed to leverage the rich fashion attributes and model the content-level transitions. Such operation was proven effective to further enhance the recommendation performance.

In the future work, we shall improve this work in the following directions. First, currently we only focus on instant intent while not considering any long-tern intent or intent in certain session. Such long-term intent affects users' behavior as well, thus should be incorporated to make the intent modeling more comprehensive. Second, this work defines the instant intent as the relationships between two adjacent items the user interacted, which is only determined by the categories of the items. However, the real-world relationships between two fashion items are more nuanced, which can be defined more sophisticated rather than limited at category level. Furthermore, the idea of simultaneously modeling user intent and the recommendation according to the intent can be applied to a wider variety of E-commence categories, rather than just fashion.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s), and do not reflect the views of National Research Foundation, Singapore. We also appreciate the fashion recognition API service provided by Visenze.

References

[1] W. Yu *et al.*, "Aesthetic-based clothing recommendation," in *Proc. Int. Conf. World Wide Web*, 2018, pp. 649–658.
[2] P. Jing, S. Ye, L. Nie, J. Liu, and Y. Su, "Low-rank regularized multi-representation learning for fashion compatibility prediction," *IEEE Trans. Multimedia*, vol. 22, no. 6, pp. 1555–1566, Jun. 2020.
[3] H. Zheng, K. Wu, J.-H. Park, W. Zhu, and J. Luo, "Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach," 2020, *arXiv:2005.12439*.
[4] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, USA, 2017, pp. 207–216.
[5] W. Chen *et al.*, "POG: Personalized outfit generation for fashion recommendation at alibaba ifashion," 2019, *arXiv:1905.01866*.
[6] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.
[7] R. He and J. McAuley, "VBPR: Visual bayesian personalized ranking from implicit feedback," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 144–150.
[8] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World wide web*, 2010, pp. 811–820.
[9] R. He, W.-C. Kang, and J. McAuley, "Translation-based recommendation," in *Proc. 11 ACM Conf. Recommender Syst.*, 2017, pp. 161–169.
[10] X. Li *et al.*, "Hierarchical fashion graph network for personalized outfit recommendation," 2020, *arXiv:2005.12566*.
[11] Y. Hu, X. Yi, and L. S. Davis, "Collaborative fashion recommendation: A functional tensor factorization approach," in *Proc. Int. Conf. Multimedia*, Brisbane Australia, 2015, pp. 129–138.
[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019.
[13] M. Quadrana, P. Cremonesi, and D. Jannach, "Sequence-aware recommender systems,"*ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018.
[14] W.-C. Kang, M. Wan, and J. McAuley, "Recommendation through mixtures of heterogeneous item relationships," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1143–1152.
[15] Â. Cardoso, F. Daolio, and S. Vargas, "Product characterisation towards personalisation: Learning attributes from unstructured data to recommend fashion products," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 80–89.
[16] X. Liu, Y. Sun, Z. Liu, and D. Lin, "Learning diverse fashion collocations via neural graph filtering," *IEEE Trans. Multimedia*, 2020, pp. 1–1, doi: 10.1109/TMM.2020.3018021.
[17] M. I. Vasileva *et al.*, "Learning type-aware embeddings for fashion compatibility," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 390–405.
[18] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "TransNFCM: Translation-based neural fashion compatibility modeling," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 403–410.
[19] W.-C. Kang, E. Kim, J. Leskovec, C. Rosenberg, and J. McAuley, "Complete the look: Scene-based complementary product recommendation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 532–10 541.
[20] G. Cucurull, P. Taslakian, and D. Vazquez, "Context-aware visual compatibility prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 617–12 626.
[21] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proc. Int. Conf. Multimedia*, Austin Texas, USA, 2017, pp. 1078–1086.
[22] Y. Lin, M. Moosaei, and H. Yang, "OutfitNet: Fashion outfit recommendation with attention-based multiple instance learning," in *Proc. The Web Conf. 2020*, 2020, pp. 77–87.
[23] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1946–1955, Aug. 2017.
[24] Y. Ding, Y. Ma, W. K. Wong, and T.-S. Chua, "Leveraging two types of global graph for sequential fashion recommendation," 2021, *arXiv:2105.07585*.
[25] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," 2019, *arXiv:1905.01997*.
[26] S. Feng *et al.*, "Personalized ranking metric embedding for next new poi recommendation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 2069–2075.
[27] P. Wang *et al.*, "Learning hierarchical representation model for nextbasket recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 403–412.
[28] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining*, Barcelona, Spain, 2016, pp. 191–200.
[29] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.
[30] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 353–362.
[31] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015.
[32] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[33] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.
[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, Cambridge, MA, USA, 2016.
[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[36] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.
[37] J. Li *et al.*, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1419–1428.
[38] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: Short-term attention/memory priority model for session-based recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1831–1839.
[39] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 17–22.
[40] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proc. Eleventh ACM Conf. Recommender Syst.*, 2017, pp. 130–137.
[41] S. Wu *et al.*, "Personal recommendation using deep recurrent neural networks in netease," in *Proc. Int. Conf. Data Eng.*, Helsinki, Finland, 2016, pp. 1218–1229.
[42] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proc. Int. Conf. Web Search Data Mining*, 2018, pp. 565–573.
[43] S. Wu *et al.*, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 346–353.
[44] Z. Wang *et al.*, "Global context enhanced graph neural networks for session-based recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 169–178.
[45] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. Int. Conf. Data Mining*, Singapore, Singapore, 2018, pp. 197–206.
[46] P. Veličković *et al.*, "Graph attention networks," 2017, *arXiv:1710.10903*.
[47] T. Chen *et al.*, "Air: Attentional intention-aware recommender systems," in *Proc. Int. Conf. Data Eng.*, Macao, China, 2019, pp. 304–315.
[48] J. McAuley, C. Targett, Q. Shi, and A. Van DenHengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 43–52.
[49] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 635–644.
[50] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.
[51] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 278–288.
[52] X. He *et al.*, "Neural collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 173–182.
[53] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. & Data Mining*, 2020, pp. 1748–1757.
[54] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.