IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 February 2023, accepted 26 March 2023, date of publication 3 April 2023, date of current version 11 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3263931

RESEARCH ARTICLE

Not All Memories Created Equal: Dynamic User Representations for Collaborative Filtering

KEREN GAIGER[1],*, OREN BARKAN[2],*, SHIR TSIPORY-SAMUEL[1], AND NOAM KOENIGSTEIN[1]

[1]Industrial Engineering Department, Tel Aviv University, Tel Aviv 69978, Israel
[2]Department of Computer Science, The Open University of Israel, Raanana 43107, Israel

Corresponding author: Noam Koenigstein (noamk@tauex.tau.ac.il)

This work was supported by the Israel Science Foundation under Grant 2243/20.

*Keren Gaiger and Oren Barkan contributed equally to this work.

ABSTRACT Collaborative filtering methods for recommender systems tend to represent users as a single static latent vector. However, user behavior and interests may dynamically change in the context of the recommended item being presented to the user. For example, in the case of movie recommendations, it is usually true that movies that the user watched more recently are more informative than movies that were watched a long time ago. However, it is possible that a particular movie from the past may become suddenly more relevant for prediction in the presence of a recommendation for its sequel movie. In response to this issue, we introduce the Attentive Item2Vec++ (AI2V++) model, a neural attentive collaborative filtering approach in which the user representation adapts dynamically in the presence of the recommended item. AI2V++ employs a novel context-target attention mechanism in order to learn and capture different characteristics of the user's historical behavior with respect to a potential recommended item. Furthermore, analysis of the neural-attentive scores allows for improved interpretability and explainability of the model. We evaluate our proposed approach on five publicly available datasets and demonstrate its superior performance in comparison to state-of-the-art baselines across multiple accuracy metrics.

INDEX TERMS Artificial neural networks, collaborative filtering, neural attention, recommender systems.

I. INTRODUCTION

Collaborative Filtering (CF) is one of the most effective and widely used methods for recommender systems [1]. Its aim is to recommend items to users based on their historical interactions with other items. As such, the underlying assumption of any CF algorithm is that users' past experiences are highly predictive of their future preferences. For instance, a user who enjoyed watching a certain type of comedy movie is likely to seek another movie that is similar to the one she previously watched. Thus, a user's memory, such as the items they have interacted with, plays a crucial role in shaping their future recommendations.

Although the role of memory in human decision-making is not fully understood, empirical evidence suggests that the human memory is dynamic, and different memories become more accessible in different contexts [2], [3], [4], [5]. Similarly, users' interests and preferences are also dynamic and context-dependent [6], [7]. According to preference construction theory, individuals do not always have a clear idea of their preferences from the outset; rather, their preferences are shaped and constructed through complex decision-making processes based on past experiences [8], [9]. Therefore, traditional CF methods that represent users as a static vector regardless of the context fail to account for the dynamic nature of users' preferences and their varying relevancy to different items.

To address this issue, there is a need to develop more sophisticated CF models that can capture the dynamic nature of users' preferences and the varying relevancy of different past experiences. These models can leverage more complex representations of users that can capture contextual information and the evolving nature of their preferences. Such models can enhance the accuracy of recommender systems and improve user satisfaction by providing recommendations that are more relevant and better personalized.

The associate editor coordinating the review of this manuscript and approving it for publication was Huiyu Zhou.

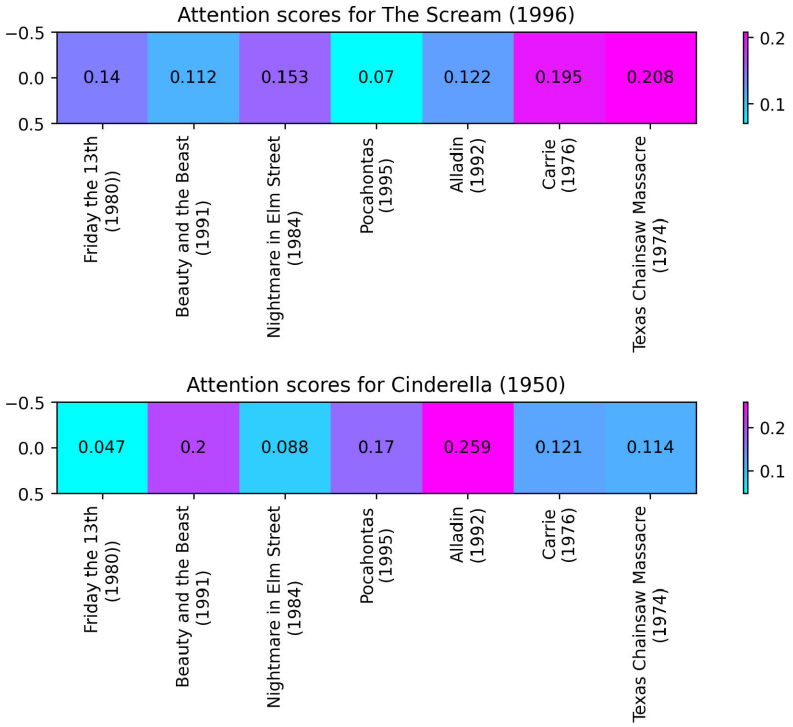


FIGURE 1. **The distribution of attention scores over a user's historical items which include both horror movies and family-fantasy movies. The scores are presented with respect to 2 different items: (1) on the upper image, with respect to the horror movie "The Scream (1996)" and (2) on the lower image, with respect to a family-fantasy movie "Cinderella (2021)". We see how the attentive relevant scores adopt with respect to the target item.**

To illustrate this gap, consider the example in Fig. 1 depicting a list of historical movies that a user has watched. The list of movies consists of both horror movies as well as some family-fantasy movies. Accordingly, the user in this example is generally interested in popular mainstream horror movies, but from time to time her 5 years old daughter joins her and they both watch an age-appropriate movie from the family-fantasy genre. Most of the time, the family-fantasy movies are irrelevant to her personal taste and interests which are mostly determined by her love of horror movies. Occasionally however, when considering a movie to watch with her daughter such as the Disney movie *Cinderella*, all the other family-fantasy movies she has watched in the past should come to the foreground and become dominant in determining her affinity to the recommended item.

This paper presents the Attentive Item-to-Vector++ (AI2V++) model, which is a novel CF model designed to capture the dynamic nature of users' preferences by adapting to changes in the relevance of historical items. The AI2V++ model is inspired by the dynamic nature of the human memory, in which different memories become more accessible in different contexts. The model uses attention mechanisms to dynamically adjust the importance of the historical items based on the item being considered for recommendation. As such, AI2V++ addresses the limitations of traditional CF models, which represent users as a static vector and ignore the varying relevancy of different past experiences. To the best of our knowledge, AI2V++ is the first CF model to explicitly account for the dynamic nature of human memory in the context of recommender systems.

The key novelty of the AI2V++ model arises from its unique multi-attentive user representation that changes and adjusts in the presence of the target item. To this end, AI2V++ employs multiple attention networks in parallel on the user's historical items with respect to a potential target item to score. Each attention network produces an attentive context-target representation, representing a different pattern in user-item affinities. This results in multiple contextualized user representations which are ''aware'' of the target item to score. At this point, the model aggregates all the context-aware user representations to form a final contextualized user representation that is subsequently used for scoring the target item. This novel *context-target* attention mechanism enables superior accuracy with respect to state-of-the-art alternative models.

AI2V++ makes another noteworthy contribution with respect to its interpretability properties. The attention mechanism embedded in AI2V++, which mimics the human brain's function, facilitates the identification of insights regarding which items the model considers as more significant for the recommendations. By highlighting relevant items in the user's history, AI2V++'s internal processes can be comprehended, and its predictions can be explicated. Consequently, AI2V++ takes a step forward in the realm of

explainability for recommender systems based on collaborative filtering.

The remainder of this paper is organized as follows: Section II covers related work concerning this research. Section III describes the proposed AI2V++ model in detail. In Section IV, we present an extensive evaluation of the AI2V++ model and compare it to state-of-the-art alternatives using multiple datasets. Finally, we summarize our findings in Section V.

II. RELATED WORK

This section provides a review of previous studies that are relevant to the present research. The literature review begins with a broad classification of collaborative filtering algorithms. Subsequently, deep learning approaches for recommender systems are discussed, with special emphasis on the use of neural attention in collaborative filtering. Finally, the topic of explainable recommender systems is briefly addressed.

A. COLLABORATIVE FILTERING

Collaborative filtering (CF) algorithms are used to model users' personalized preferences using historical user-item interactions [1]. CF models are characterized as either *explicit* e.g., item ratings, thumb up/down, or *implicit* e.g., clicks, purchases, etc. Initial research, following the Netflix Prize competition [10] focused mainly on explicit data. However, due to the lack of explicit data in industrial applications, research efforts shifted to focus on implicit models [11], [12].

Implicit user feedback can be challenging due to the ambiguity of interpreting 'non-observed' interactions. Hence, *point-wise* and *pairwise* methods were proposed to alleviate this challenge and learn latent user representations. In pairwise methods e.g., [13] the positive user-item interaction is contrasted with another item that the user did not interact with. In point-wise methods e.g., [11], a positive user-item interaction is contrasted against all other items that the user did not interact with sometimes via sampling.

A different approach to implicit feedback collaborative filtering is to learn only item representations without user representations i.e., *implicit user* learning. These methods emphasize the importance of learning item-to-item semantics rather than user-to-item predictions. For example, [14] proposed learning item representations from implicit feedback in a Euclidean space. The I2V model [15] is a popular method for learning *static* item representations based on CF item co-occurrences [15]. The AI2V++ model in this paper belongs to this category. It is inspired by I2V and adds to it the ability to dynamically build implicit user representations based on their items.

The I2V model is a well-known CF technique that utilizes item co-occurrences to acquire hidden item representations. I2V is based on the Skip-Gram with Negative Sampling (SGNS) approach, which is also used in Mikolov et al.'s influential Word2Vec model to learn semantic word representations [16]. In recent years, several studies have demonstrated the usefulness of incorporating neural attention mechanisms to static word embeddings, enabling them to capture the dynamic qualities of words in relation to their context [17], [18]. Following this trend, the authors of this paper introduced the AI2V recommendation model, which enhances the I2V model by presenting a novel cross-attention mechanism that modifies user representations in response to the item being rated [19].

The model in this paper, dubbed AI2V++, enhances our earlier conference presentation of the AI2V model [19]. Two key modifications were made to the original AI2V algorithm. First, we introduced ordinal information into the context-target attention mechanism by hierarchically learning global and personal ordinal biases. It should be noted that the hierarchical mechanism integrated into AI2V++ represents a novel and distinct approach compared to the typical method of learning positional embeddings that is commonly employed in most transformers, as seen in previous works such as [17] or in [20]. Second, we changed the categorical cross-entropy loss in AI2V to a binary cross-entropy loss, which is more suitable for multi-label problems, as there are typically multiple appropriate items per user in recommender systems. Our experimental results demonstrate that both modifications significantly enhance the original AI2V model. In particular, the incorporation of ordinal information enables AI2V++ to consider the order in which a user's items were consumed, resulting in a considerable performance improvement over AI2V. Additionally, we illustrate how attentive score analysis in AI2V++ can be utilized for explainability and interpretability. The code for AI2V++ was made available on GitHub[1] and is expected to be easily accessible for researchers and practitioners alike.

B. DEEP LEARNING FOR COLLABORATIVE FILTERING

In recent years, numerous deep learning-based recommender system models have been proposed, as documented in [21], [22], [23], and [24]. A particular strand of research aims to substitute the conventional inner-product operation found in Matrix Factorization (MF) models with deep neural networks. For instance, AutoRec [25] uses autoencoders to predict ratings, while Neural Collaborative Filtering (NCF) [26] estimates user-item interactions through Multi-Layer Perceptions (MLP). The value of this approach is currently being debated within the research community, with some studies suggesting that an inner product may suffice for CF tasks and that the added complexity may be unnecessary [27], [28]. Although the primary focus of this paper is not on this topic, the AI2V++ model does utilize neural scoring, which has been shown to enhance predictions in four of the five evaluation datasets analyzed.

Another line of work seeks to employ Graph Neural Networks (GNNs) for CF. An example of this is the Neural Graph Collaborative Filtering (NGCF) model [29], which uses the well-established GCN model [30], [31], originally

[1] https://github.com/kerengaiger/ai2v

designed for graph classification, to perform CF. In contrast to conventional approaches that train distinct embeddings for users and items, NGCF learns a function that creates embeddings by collecting and aggregating features from a user's local neighborhood. NGCF has become a state-of-the-art model for CF by achieving superior performance compared to many notable models, including BPR [13], CMN [32], HOP-Rec [33], PinSage [34], and GC-MC [35]. Recently, LightGCN was introduced as a simplified version of NGCN by retaining only the most relevant components of NGCN. LightGCN is simpler to implement and train, yet it has been shown to outperform NGCN [36].

A further line of research has focused on the integration of neural attention mechanisms. The present study falls within this category. Attention mechanisms have emerged as a crucial component in numerous deep learning models [37]. Specifically, self-attention and transformer models have yielded exceptional outcomes in various Natural Langauge Processing (NLP) tasks, such as language translation and understanding [17], [20]. The success of self-attention models in NLP has triggered widespread adaptation of these models for computer vision [38], [39], [40]. Moreover, the versatility and scalability of transformer models have enabled the processing of multiple modalities (e.g., text and images) using similar processing blocks [41].

Incorporating attention mechanisms into recommender system models has been an active area of research. Attention-based recommendation models mostly utilize self-attention. One such model is presented in [42], which uses two transformer encoders to capture mobile user click behavior. Another example is SASRec [43], which employs a self-attention mechanism to represent each item in a user's item sequence and generates a user representation based on the final attention block. The user representation is multiplied by the target item embedding vector to produce an affinity score. To emphasize the importance of the last item in the sequence, SASRec also includes a residual connection between the non-contextualized representation of the last item and the final user representation. SASRec has been demonstrated to outperform several popular algorithms, including BPR [13], FPMC [44], TransRec [45], GRU4Rec [46], GRU4Rec+ [47] and Caser [48]. Finally, Bert4Rec [49] is another model for sequential recommendations which is closely related to SASRec. Bert4Rec also employs self-attention but instead of one-directional attention, it employs bi-directional attention via the Cloze task [50].

AI2V++ differs from these models in several aspects: First, in contrast to the above models, which are focused on sequential recommendations, AI2V++ is a traditional CF recommendation model that performs a different prediction task and is evaluated using different datasets. Importantly, its approach to user representation also sets AI2V++ apart. AI2V++ assumes that users' interests are dynamic and can change in response to different target items. To capture this, AI2V++ uses context-target attention to dynamically adjust the user's representation based on the presence of the target item and does not employ self-attention, as do all the aforementioned models. This approach is inspired by how the human brain works, where different parts of memory become relevant in different contexts.

In general, the AI2V++ model distinguishes itself from any transformer-based model in two ways. Firstly, AI2V++ does not utilize self-attention and instead employs cross-attention on the target item. Second, AI2V++ relies on cosine similarity for its attention mechanism. Furthermore, AI2V++ employs a compound neural scoring function to compute the similarity between the user representation and the target item, rather than a simple inner product. Finally, while transformer-based models leverage positional encoding to encode item sequence information, AI2V++ uses a novel set of hierarchical ordinal bias scalars within the attention layer to learn the relevant order of items in a user's history.

He et al. introduced NAIS, Neural Attentive Item Similarity model (NAIS) for item-based CF from [51]. Similar to AI2V++, NAIS employs cross-attention in order to learn the relative importance of the historical items in a user's profile with respect to the prediction. In that respect, NAIS is arguably the most similar model to AI2V++. However, there are several key differences between NAIS and the model in this paper: (1) NAIS does not learn user representations. Instead, it is an item-centric approach in which the predictions are computed according to the relation of the target item and each of the user's historical items without ever computing user representations explicitly. (2) NAIS mostly uses the item embeddings directly, or in one of its versions (design 3 in [51]), NAIS learns a single projection matrix on the item embeddings in order to compute the attention scores (Equation 7 in [51]). In contrast, AI2V++ employs 4 different types of projections on the item embeddings: 2 projections for the context items and 2 for the target items. In each case, context or target, the first projection is used in order to transfer the item into the attention scoring space while the second projection is used in order to compute the dynamic user representation and the target item representation prior to the final user-item scoring function. By employing different projections for the attention and for the final prediction, AI2V++ is able to *disconnect* these two distinct functions, which gives it much more flexibility. For example, consider the case of items in the user history that are in general very similar to the target item, yet their relative importance (attention) with respect to the user's taste is marginal. (3) Another key difference is the fact that AI2V++ injects ordinal information into the context-target attention mechanism, which as we show in Section IV-F, makes a dramatic contribution to the model's accuracy. This is achieved by a novel learnable mechanism of hierarchical global and personal ordinal biases which help emphasize recent events over older events. (4) While NIAS employs a single cross-attention unit, AI2V++ employs multi-head attention which gives it further descriptive power. (5) Last but not least, a key advantage of AI2V++ stems from its ability to extract

TABLE 1. AI2V++ notation summary.

x	A user represented as a timely ordered sequence of items.
x_{j-1}	A sub-user represented as a sequence of the first $j-1$ items.
$\mathbf{u}_{l_i}$	A context item representation for item l_i.
$\mathbf{v}_{l_i}$	A target item representation for item l_i.
$\mathbf{A}_c$	Context items projection matrix for the attention keys.
$\mathbf{B}_c$	Context items projection matrix for the attention values.
$\mathbf{A}_t$	Target item projection matrix for the attention query.
$\mathbf{B}_t$	Target item projection matrix for the final prediction head.
$\mathbf{a}_{j-1}$	An attentive representation of sub-user x_{j-1}.
α_{jm}	An attention score for item l_m with respect to target item l_j.
λ_i	A *global* ordinal bias for the positional importance of the i-th position (reverse order).
λ_i^x	User's x *personal* ordinal bias at position i.
$\mathbf{z}_{j-1}$	A multi-attentive representation for sub-user x_{j-1}.
b_{l_j}	A popularity bias for item l_j.

intuitive explanations for its predictions. Explainable AI algorithms and in particular explaining recommender systems is an important open research question [52]. While it may be possible to extract meaningful explanations from NAIS, the authors in [51] do not address the issue at all. In contrast, the current paper explains and demonstrates how AI2V++ provides interpretability over its predictions which can be further harnessed for generating user-intuitive explanations.

C. EXPLAINABLE RECOMMENDER SYSTEMS

In recent years, the need for explainable AI has become a topic of increasing interest and importance for both the research community and industry, as evidenced by the growing number of publications and regulations in the field [53], [54], [55]. For example, the European Union General Data Protection Regulation determines that users have a basic "right to an explanation" concerning algorithmic decisions based on their personal information [55]. Similar regulations either exist or are publicly proposed in other countries. Specifically, in the context of recommender systems, the goal of explanations is to provide justifications for recommendations in a way that is understandable to human users [52]. Previous research has shown that transparency and interpretability are crucial for building trust in recommender systems and ensuring their effectiveness [56]. In this regard, AI2V++ offers a distinct advantage over many existing CF algorithms. By analyzing AI2V++'s attention scores over the user's historical items, the model's inner workings can be revealed, providing transparency and interpretability that is often lacking in other CF approaches that operate as a "black box" to end users.

III. THE MODEL

The AI2V++ model builds upon the I2V model [15] and adds to it the ability to learn dynamically adaptive user representations. Hence, we first formalize and provide a brief overview of the Item2Vec (I2V) model which serves as the basis for AI2V++. Then, we continue to describe the AI2V++ model in detail.

A. ITEM2VEC (I2V)

Let $\mathcal{I} = \{i\}_{i=1}^{M}$ be a set of M item identifiers. For each item i, I2V learns latent context and target vectors $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^d$. These latent vectors are estimated via implicit factorization of the items' co-occurrence matrix. Specifically, the training data for I2V consists of a list $x = (l_1, \ldots, l_K)$ for each user of the historical items that were co-consumed by that user.

Without loss of generality, we consider a dataset of a single user x, where the extension to multiple users is straightforward and can be found in [15]. The objective of I2V is to learn item co-occurrences. This is achieved by minimizing the following loss function:

$$\mathcal{L}_x^{I2V} = -\sum_{i=1}^{K}\sum_{j<i}^{K} \log p(l_j|l_i), \tag{1}$$

with

$$p(l_j|l_i) = \sigma(s(l_i, l_j)) \prod_{k\in\mathcal{N}} \sigma(-s(l_i, k)), \tag{2}$$

where $s(i,j) = \mathbf{u}_i^T \mathbf{v}_j$, $\sigma(x) = (1+\exp(-x))^{-1}$ and $\mathcal{N} \subset I$ is a subset of items that are sampled from I according to the unigram item popularity distribution raised to the power of 0.5. The items in $\mathcal{N}$ are treated as negative context items with respect to the target (positive) item l_j.

In order to mitigate the popularity bias in common CF datasets, I2V further applies a subsampling procedure in which positive items are randomly discarded from users according to their popularity. The amount of subsampling is controlled by a hyperparameter that is adjusted w.r.t. the dataset statistics as explained in [15].

During the training phase, I2V learns the sets of context and target vectors $\mathbf{U}, \mathbf{V} \subset \mathbb{R}^d$ by minimizing $\mathcal{L}_x^{I2V}$ using any stochastic gradient descent method. In the inference phase, the affinity between the context and target items i and j is based on the cosine similarity as follows:

$$\cos(\mathbf{u}_i, \mathbf{v}_j) = \frac{\mathbf{u}_i^T \mathbf{v}_j}{|\mathbf{u}_i||\mathbf{v}_j|}. \tag{3}$$

The I2V model is commonly used in the recommender systems community especially for learning item similarities from collaborative filtering data. In what follows, we keep the notations for the context and target vectors from above, i.e., $\mathbf{U}, \mathbf{V} \subset \mathbb{R}^d$.

B. ITEM2VEC++ (AI2V++)

The AI2V++ model is designed to estimate the likelihood of a user consuming a target item based on her past consumption history. Compared to the I2V model, AI2V++ incorporates several modifications. First, it utilizes a novel attention mechanism that allows personalization by selectively attending to the user's historical items in the context of the target item. Second, AI2V++ introduces a hierarchical ordinal bias scalar to the attention layer, which enables the model to learn the relevant order of items within the user history. Additionally, AI2V++ employs a neural scoring function to compute the similarity between the user representation and candidate items, instead of the dot-product function used in I2V. Finally, the model incorporates target biases to address popularity

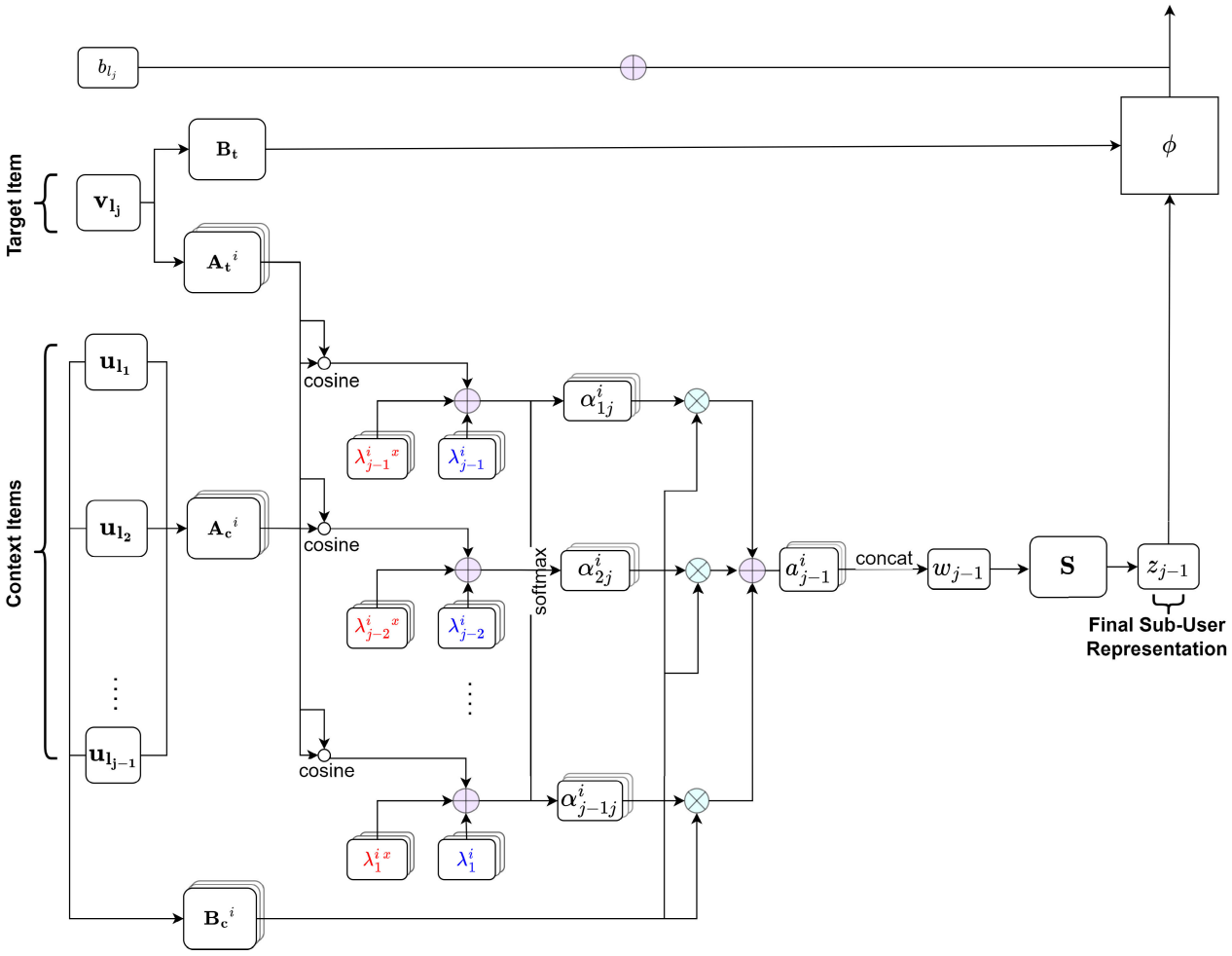


FIGURE 2. A schematic illustration of the AI2V++ model. First, the context and target item embeddings are multiplied by the learnable transformation matrices $\mathbf{A}_c^i$ and $\mathbf{A}_t^i$, respectively. Then, cosine similarity is calculated between the transformed target item and each of the transformed context items. The attention weights α_{mj}^i are given by the softmax operation applied on the sum of the cosine similarities, the global ordinal biases λ_m, and the user-personal ordinal biases λ_m^x. The sub-user representation $\mathbf{a}_{j-1}^i$ is the sum of the transformed context items weighted by the attention weights. This calculation is repeated over N multiple attention heads, where i denotes the attention head index. The final sub-user representation $\mathbf{z}_{j-1}$ is given by concatenating $\{\mathbf{a}_{j-1}^i\}_{i=1}^N$ and passing through $\mathbf{S}$. In parallel, the target item representation is passed through $\mathbf{B_t}$ and scored according to the neural scoring functions ϕ Eq. 7 and ω Eq. 8. Finally, the popularity bias b_{l_j} of the target item is added to account for general popularity patterns.

biases and avoid the need for subsampling as used in I2V. The multi-attentive context-target mechanism of AI2V++ is illustrated in Figure Fig. 2. The details of the model are explained in a step-by-step manner, and the notations used in the model are summarized in Tab. 1 for ease of reference.

1) A DYNAMIC USER REPRESENTATION

Consider a user with a list $x = (l_1, \ldots, l_k)$ of historical items which are ordered by the time of consumption. We denote a sub-user by $x_{j-1} = (l_1, \ldots, l_{j-1}), (j < k)$, i.e., a sub-user is simply a sub-sequence of the historical items consumed by the user. The attentive context-target mechanism produces an attentive sub-user representation for x_{j-1} as follows:

$$\mathbf{a}_{j-1} = \sum_{m=1}^{j-1} \alpha_{jm} \mathbf{B}_c \mathbf{u}_{l_m}, \tag{4}$$

where $\mathbf{u}_{l_m} \in \mathbb{R}^d$ is the item representations for item l_m, $\mathbf{B}_c \in R^{d \times d}$ is a learnable linear mapping that maps the historical context item vectors to a new space, and α_{jm} are the attention weights. These attention weights are computed dynamically based on the target item l_j according to:

$$\alpha_{jm} = \frac{\exp(\tau \cos(\mathbf{A}_c \mathbf{u}_{l_m}, \mathbf{A}_t \mathbf{v}_{l_j}) + \lambda_{j-m} + \lambda_{j-m}^x)}{\sum_{n=1}^{j-1} \exp(\tau \cos(\mathbf{A}_c \mathbf{u}_{l_n}, \mathbf{A}_t \mathbf{v}_{l_j}) + \lambda_{j-n} + \lambda_{j-n}^x)}, \tag{5}$$

where $\mathbf{v}_{l_j} \in \mathbb{R}^d$ is the target item's vector, and $\mathbf{A}_c, \mathbf{A}_t \in R^{d \times d_\alpha}$ are learnable linear mappings from the original context and target spaces to a d_α-dimensional context and target attention space, respectively. The attention scores are based on the cosine similarity within this attention space. The hyperparameter τ controls the attention sphere's radius.

The AI2V++ model includes global and personal ordinal biases that incorporate temporal information: $\Lambda = \{\lambda_i\}_{i=1}^{C_{max}}$

are *global* (shared by all users) ordinal biases that learn the importance of each position in the context items sequence (C_{max} is the maximal sequence length of a sub-user), and $\Lambda^x = \{\lambda_i^x\}_{i=1}^{C_{max}}$ are the *personal* ordinal biases for user x that enables personalized correction to the global ordinal biases Λ. Λ^x allows for a subsequent layer of personalization that is instrumental for users that do not conform to the global temporal trend captured by the global ordinal biases Λ.

During optimization, we employ L2 regularization on Λ^x s.t. their effect is pronounced only if the specific user behavior justifies a different pattern than the global trend learned by Λ (see Eq. 10). In the evaluation of the model (Sec. IV, we show that the addition of the global and personal ordinal biases, which were not included in the conference presentation of AI2V [19], significantly contributes to the model's accuracy.

Finally, the attentive sub-user representation $\mathbf{a}_{j-1}$ (Eq. 4) is a convex combination of her historical item representations. $\mathbf{a}_{j-1}$ is dynamic and depends on the item l_j: the importance and hence the weight of each historical item l_m in x_{j-1} is governed by its affinity α_{jm} to the target item l_j. In other words, if l_j changes, $\mathbf{a}_{j-1}$ changes as well. This mechanism tries to mimic how the human mind works: when considering different items, different memories come to the foreground and influence our opinion.

2) A MULTI-ATTENTIVE USER REPRESENTATION

AI2V++ applies the above attentive context-target mechanism multiple times in order to learn several user representations in parallel. These representations are then aggregated to form a *multi-attentive* user representation. Specifically, we propose to learn N context-target attention mechanisms, in parallel. Each attention mechanism is associated with different sets of learnable parameters $\{\mathbf{A}_c^i, \mathbf{A}_t^i, \mathbf{B}_c^i, \Lambda_i, \{\Lambda_i^x\}_{x=1}^X\}_{i=1}^N$, where X is the number of users. This process produces N attentive context-target representations for each user $\{\mathbf{a}_{j-1}^i\}_{i=1}^N$ according to Eq. 4. Then, the final multi-attentive user representation $\mathbf{z}_{j-1}$ is given by:

$$\mathbf{z}_{j-1} = \mathbf{S}\mathbf{w}_{j-1}, \tag{6}$$

where $\mathbf{w}_{j-1} = [(\mathbf{a}_{j-1}^1)^T, \ldots, (\mathbf{a}_{j-1}^N)^T]^T$ is a stacked matrix based on the N attentive context-target representations, and $\mathbf{S} \in \mathbb{R}^{d \times Nd}$ is a learnable linear mapping that transforms the multiple user representations of the sub-user back to the original dimension. This allows AI2V++ to learn various types of attention functions and aggregate the information extracted by each attention function into the final multi-attentive user representation.

3) THE AI2V++ SIMILARITY FUNCTION

The multi-attentive user representation $\mathbf{z}_{j-1}$ from Eq. 6 encodes the output from the multiple attentive context-target units. AI2V++ computes the affinity between a user and a target item by applying a neural scoring function $\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ as follows:

$$\phi(\mathbf{u}, \mathbf{v}) = \mathbf{W}_1 \text{ReLU}(\mathbf{W}_0([\mathbf{u}, \mathbf{v}, \mathbf{u} \circ \mathbf{v}, |\mathbf{u} - \mathbf{v}|])), \tag{7}$$

where $\circ$ denotes the Hadamard product, and $\mathbf{W}_0 \in \mathbb{R}^{d \times 4d}$ and $\mathbf{W}_1 \in \mathbb{R}^{1 \times d}$ are learnable linear mappings (matrixes). It is a neural network with a single ReLU-activated hidden layer and a scalar output. According to our experiments, this scoring function, inspired by [57], outperformed the use of the dot-product as the similarity function. The final score of sub-user x_{j-1} and the target item l_j is given by:

$$\omega(x_{j-1}, l_j) = \phi(\mathbf{z}_{j-1}, \mathbf{B}_t \mathbf{v}_{l_j}) + b_{l_j}, \tag{8}$$

where ϕ is the neural scoring function from Eq. 7, $\mathbf{B}_t \in R^{d \times d}$ is a learnable linear mapping, and b_{l_j} is a popularity bias for the target item l_j.

4) THE LOSS FUNCTION

Our goal is to compute the probability of the item l_j given the historical items in x_{j-1}, i.e., $p(l_j|x_{j-1})$. To this end, AI2V++ models $p(l_j|x_{j-1})$ according to:

$$p(l_j|x_{j-1}) = \sigma(\omega(x_{j-1}, l_j)) \prod_{k \in \mathcal{N}} \sigma(-\omega(x_{j-1}, k)), \tag{9}$$

where $\omega(\cdot, \cdot)$ is the AI2V++ score function from Eq. 8, and $\mathcal{N}$ is defined in the same manner as in Eq. 2. Finally, AI2V++ loss for a user x is given by:

$$\mathcal{L}_x = -\sum_{j=2}^{K} \log p(l_j|x_{j-1}) + \gamma \sum_{i=1}^{N} \|\Lambda_i^x\|_2^2, \tag{10}$$

where γ is a hyperparameter that controls the regularization of the personal ordinal biases (correction) for user x.

The optimization proceeds with stochastic gradient descent on the BCE loss with negative sampling. At the inference phase, the similarity between a user x and a target item k is computed by $\omega(x, k)$.

IV. EVALUATIONS

In this section, we describe the experimental setup and the results of our evaluations.

A. EXPERIMENTAL SETUP

Training, validation, and test sets were generated using the leave-one-out approach, i.e., for a user with K items, we allocated the K'th item (the last item) for the test set, and the item before it (item $K-1$) for the validation set. The rest of the items (items 1 till $K-2$) were used for the training set. Since not all sub-user sequences are of the same length, we fixed a window size based on the longest sequence and padded shorter sequences at their beginning accordingly. Furthermore, when training the model, we set the attention weights of those padded positions to zero so they won't affect the sub-user representation. The prediction of the hidden item for each user was done by applying the similarity function from Eq. 7 to all candidate items and returning the item with the maximal score.

TABLE 2. Dataset statistics.

Database	Users	Items	Ratings	Sparsity %
MovieLens-1M	5,765	1,865	220,311	97.95
Netflix	10,677	2,121	396,316	97.92
Yahoo!	19,151	17,711	85,147	98.25
Moviesdat	11,142	2,949	1,670,318	99.69
Amazon Books	31,202	2,111	305,156	99.54

B. DATASETS

We consider several datasets from different domains. Each of the datasets consists of the following fields: user ID, item ID, rating, and timestamp. On each dataset, we filtered users with less than 4 items or more than 1,000 items. AI2V++ is designed to employ implicit ratings, hence explicit numerical ratings were first scaled to a 5-star rating scale, and then ratings of 4 stars and above were considered as positive examples. The following datasets were considered:

- **MovieLens 1M:** The MovieLens-1M database [58] has been widely used to evaluate collaborative filtering algorithms [59]. It consists of 1 million ratings from 6,040 users to 3,883 movies.
- **Moviesdat:** The Moviesdat dataset [60] consists of 26 million ratings from 270,000 users to 45,000 movies.
- **Netflix:** The Netflix dataset [10] consists of more than 100 million ratings by 480,189 users to 17,770 movies.
- **Yahoo! Music:** From the Yahoo! Music dataset [12] we sampled 19,989 users with 30,000 items and around 3.36 million user ratings.
- **Amazon Books:** The Amazon Books dataset [61] is based on book reviews crawled from amazon.com. This dataset consists of 22.5 million reviews given by 8.9 million users to 2.37 million books.

Before describing the evaluation process, we briefly investigate some relevant statistical properties of the datasets above. We measure the sparsity of the datasets by calculating the percentage of user-item pairs without ratings out of the entire user $\times$ item ratings matrix. As expected for collaborative filtering datasets, the sparsity level was high in all cases. In particular, rating sparsity in the Moviesdat and Amazon datasets is significantly higher. Table 2 summarizes these statistical properties for each dataset. In addition, Fig. 3 presents the popularity distribution of the different datasets used in this research on a logarithmic scale. As can be seen, all datasets present very skewed distributions. However, the items in the Yahoo! Music dataset suffer from a much higher degree of popularity skew.

C. EVALUATION METRICS

The qualitative measurements in this paper follow [59], [62] and cover the following metrics:

- **Hit Ratio at K (HR@K):** The percentage of the predictions made by the model, where the positive test item was found in the top K items suggested by the model. Formally, a test-set tuple $(x_{1:t-1}, l_t)$ is scored '1' if the test item l_t was ranked in the top K recommendations produced by the model w.r.t. to the user $x_{1:t-1}$ and '0' otherwise:

$$\text{HR@K} = \begin{cases} 1, & \text{if } pos_t \leq K \\ 0, & \text{otherwise,} \end{cases}$$

where pos_t is the position of the test item in a ranked list of all items. We report the mean HR@K for all the users in the dataset. Note that, unlike the other metrics, the HR@K measure ignores the exact position of the test item as long as it appears in the top K.

- **Mean Reciprocal Rank at K (MRR@K):** This metric reports the average Reciprocal Rank at K (RR@K), where the reciprocal rank is set to zero if the target item does not appear in the top K recommendations:

$$\text{RR@K} = \begin{cases} \dfrac{1}{pos_t}, & \text{if } pos_t \leq K \\ 0, & \text{otherwise,} \end{cases}$$

where pos_t is the position of the target item within the ranked list of items for the user. We report the mean RR@K for all the users in the dataset.

- **Normalized Discounted Cumulative Gain at K (NDCG@K):** This metric reports the Discounted Cumulative Gain at K (DCG@K) normalized by the Ideal Discounted Cumulative Gain at K (IDCG@K) which is achieved by the optimal ranking order. In our case, since there is only one test item, the NDCG@K is calculated as follows:

$$\text{NDCG@K} = \begin{cases} \dfrac{1}{\log_2(pos_t + 1)}, & \text{if } pos_t \leq K \\ 0, & \text{otherwise,} \end{cases}$$

where pos_t is the position of the test item. We report the mean NDCG@K for all the users in the dataset.

- **Mean Percentage Rank (MPR):** This metric is a recall-oriented metric that is used to measure the average user satisfaction with items in an ordered list. MPR considers the entire list of ranked items (not just the top K). The percentile rank is defined as follows:

$$\text{PR} = \frac{pos_t}{N}, \tag{11}$$

where pos_t is the position of the target item within the ordered list of all items for the user, and N is the number of items in the catalog. The Mean Percentage Rank is the mean PR_u for all the users in the dataset. Note that, unlike the previous metrics, lower values of MPR are more desirable, as they indicate that the test item was ranked closer to the top of the recommendation lists.

D. BASELINES

The following baselines were considered for evaluation:

- **Popularity (POP):** This simple baseline ranks the items based on their popularity and recommends the most

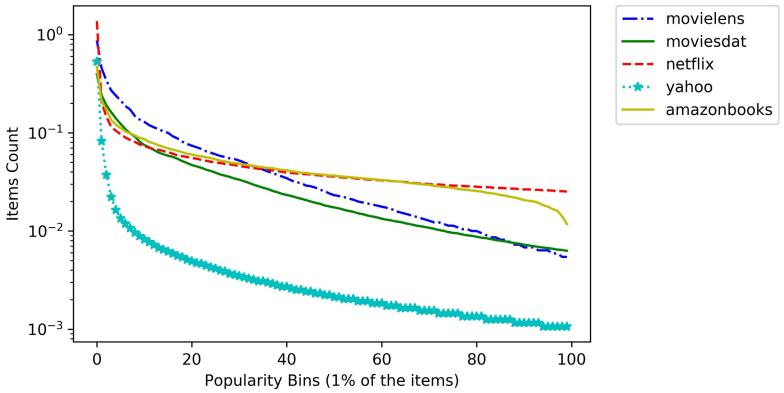


FIGURE 3. Item popularity skew for the different datasets - from the most popular item (first percentile on the left) to the least popular item (on the right). We see that all datasets suffer from a strong popularity skew and a long tail of less popular items. In the Yahoo! Music dataset this phenomenon is even more prominent.

popular items to all users. While this approach lacks personalization, it was shown to perform well on many collaborative filtering tasks [59].

- **Item2Vec (I2V):** The Item2Vec (I2V) model is an item-based Collaborative Filtering model that gained much popularity in recent years [15]. As explained earlier, AI2V++ generalizes I2V by incorporating a neural attention mechanism in order to dynamically generate a user representation. As such, I2V serves as an ablated version of AI2V++ that showcases the contribution of AI2V++'s improvements.
- **Neural Collaborative Filtering (NCF):** Neural Collaborative Filtering (NCF) [26] employs Generalized Matrix Factorization (GMF) [63] with a multi-layer perceptron (MLP) for the user-item interaction function. The NCF model showed significant improvements over many well-know state-of-the-art methods such as ItemKNN [64], BPR [13], eALS [65], and WMF [11].
- **LightGCN:** LightGCN [36] is the leading graph-based model for Collaborative Filtering [30], [31]. The model learns user and item embeddings by linearly propagating them on the user-item interaction graph, and a weighted sum of the embeddings from all layers is used as the final user representation. LightGCN is an improvement over NGCF [29] which was shown to outperform many previous models such as graph-based GC-MC [35] and PinSage [34], neural network-based models such as NCF [26] and CMN [32], and factorization-based models such as BPR [13] and Hop-Rec [33]. In addition, in [36], LightGCN was shown to outperform Mult-VAE [66], and GRMF [67].
- **SASREC:** Self-attentive sequential recommendation (SASRec) [43] incorporates a self-attention mechanism to utilize user 'context' activities based on actions they have performed recently. SASRec was shown to outperform strong baselines such as GRU4Rec [46], GRU4Rec+ [47], and Caser [48].
- **NAIS:** Neural Attentive Item Similarity model (NAIS) for item-based CF from [51]. Similar to AI2V++, NAIS employs cross-attention based on the target item but differs from AI2V++ in multiple aspects as described in Section II.
- **AI2V++:** The model presented in this paper.
- **AI2V-Vanilla:** This is the basic version of AI2V from our conference presentation [19]. The differences between AI2V and AI2++ were covered in Section II. AI2V-Vanilla is an ablated version t showcases the relative improvements made in AI2V++.
- **AI2V-Pos:** An ablated version of AI2V++ which is based on AI2V-Vanilla plus the hierarchical ordinal biases (from Sec. III-B). This version comes to showcase the specific contribution of the ordinal biases to AI2V++.
- **AI2V++ dot:** An ablated version of AI2V++, in which we replaced of the neural scoring function from Eq. 7 with the dot-product. This ablated version comes to showcase the relative contribution of AI2V++'s neural scoring function.

E. HYPER-PARAMETERS AND CONFIGURATION

Hyper-parameters tuning was performed on all baselines and ablated versions using the Optuna optimization framework [68] on the validation set. Specifically, AI2V++'s hyper-parameters used in this paper are as follows: The dimensions of the attention layer weights, A_t, A_c, and B_c, were set to 50. The parameter controlling the radius sphere in the attention weights calculation, τ, was set to 1. The number of negative items that were sampled for each positive target item was set to 7. The model was trained using Adagrad [69] and a mini-batch of 32 samples until the learning process has finished and overfitting has been achieved. Then, the model with the best validation score was chosen for evaluation. The following hyper-parameters were optimized separately for each dataset (using the validation set): learning rate, embedding size, and the number of attention heads.

F. RESULTS

The current study presents a comprehensive evaluation of the AI2V++ model. The evaluation is divided into three main sections. First, in Sec.IV-F1, we report the quantitative accuracy results for various models, datasets, and evaluation metrics, and analyze the results based on item popularity. Second, in Sec.IV-F2, we provide a detailed investigation of the role of ordinal biases in the AI2V++ model. Finally, in Sec. IV-F3, we illustrate the dynamic construction of user representations by AI2V++ and demonstrate how attention scores can be utilized to provide model interpretability and explainability.

1) COMPETATIVE RESULTS

Tables 3-7 summarize the extensive evaluations on the MovieLens-1M, Netflix, Yahoo!, Amazon Books, and Moviesdat datasets respectively ($p \leq 0.05$). W The superiority of the AI2V++ model is clearly noticeable across the different datasets and metrics. In all cases, the ordinal biases, which are part of the contributions made in the current paper, significantly improve the results over the AI2V-Vanilla. Additionally, in most cases employing the BCE loss, which is another contribution of the current paper, yields better results than the original CCE loss.

Next, we turn to analyze the results as a function of item popularity. It has been shown that in collaborative filtering problems, much of the signal lies in simple popularity biases [70]. For example, the winning model in the Netflix Prize competition [10] managed to explain 42.6% of the ratings' variance i.e., $R^2 = 42.6\%$, but the vast majority of the learned signal was attributed to popularity biases which explained a whopping $R^2 = 32.5\%$ of the variance (without any personalization) [71].

Following this insight, we wish to investigate the model's results as the effect of popular items is artificially diminished. To this end, we gradually remove popular items from the dataset and evaluate the results on the remaining, less popular, items. Figures 4a-4e depict the HR@20 metric (y-axis) after removing the most popular items (x-axis) for each dataset. We see that the HR metric monotonically decreases as more popular items are removed. This is expected since the popular items are easier to predict. Importantly, we notice a much milder decrease in the AI2V++ variants compared to the baselines. By being able to dynamically focus on different items in the user's history, AI2V++ can turn its focus to any specific item in the user's history, even if that item does not agree with the general or recent user's taste. This gives the AI2V++ variants an advantage in recommending long-tail items, a highly important property for recommender systems [59], [72]. In contrast, when the user representation is static, it tends to be more focused on the popular items and performs poorly in the presence of less popular items. Note that the moderate decrease in the Moviesdat and Amazon Books datasets is attributed to the higher inherent sparsity of these two datasets compared to the other ones (as seen in Table 2), however, the general trend remains - the AI2V++ variants maintain superiority in the long-tail.

2) A DEEPER ANALYSIS INTO ORDINAL BIASES

From Tables 3-7 we learn that the additional ordinal biases in AI2V++ are responsible for a significant improvement in the model's accuracy. This result is expected as user preferences are known to be drifting over time and temporal dynamics need to be addressed [73]. As explained earlier, the addition of ordinal biases enables the AI2V++ model to be aware of the order in which the user consumed the items. The global ordinal biases $\Lambda = \{\lambda_i\}_{i=1}^{C_{max}}$ attribute different learnable weights according to the consumption order. In addition, the personal ordinal biases $\Lambda^x = \{\lambda_i^x\}_{i=1}^{C_{max}}$ enable per-user personalized correction to the global ordinal biases Λ.

Considering the notable contribution of this component, we wish to better understand the effect of the ordinal biases in the AI2V++ model. Fig. 5 depicts the learned global ordinal bias values of the last twenty items in a user's sequence. In order to present model parameters from different model instances (i.e., different datasets) on a single scale, the weights were normalized by the weight of the highest bias value. A common trend in all datasets is the gradual decrease in items' importance where the most recent items are emphasized over less recent ones.

By comparing the trends of different datasets, further interesting insights can be extracted: For example, we learn that in the Yahoo! Music dataset, the decrease in item importance is considerably more significant than in the other datasets. This implies that in this dataset the non-stationary temporal trends are of higher importance, in accordance with a deeper analysis performed on this dataset for KDD-Cup'11 [71]. In contrast, the non-stationary temporal effects on the Netflix dataset, while evidently significant, are of less importance than in other datasets.

3) INTERPRETABILITY WITH AI2V++

In what follows, we present an attention score analysis that demonstrates interpretability by exposing the inner workings of the model. Figure 6 presents visualizations of the attention weights for users from MovieLens-1M. The attention scores are calculated when scoring the last movie in the sequence (the test item) for an AI2V++ model with a single attention head. The first example in Fig. 6 relates to the movie "Clear and Present Danger (1994)". Of the user's train items, the highest score was given to the movie "Patriot Games (1992)". Both movies are action thriller films, based on books by Tom Clancy (a novelist), with Harrison Ford (an actor) as Jack Ryan (the lead character). In fact, "Clear and Present Danger (1994)" is a sequel to "Patriot Games (1992)". Other movies with high scores such as "In the Line of Fire (1993)", "Fugitive (1993)", and "The Hunt for Red October (1990)" are all related action thrillers. In fact, "The Hunt for Red October (1990)" is the first movie in Clancy's Jack Ryan movie series.

TABLE 3. Evaluation results on the MovieLens-1M dataset.

Model	HR@K			MRR@K			NDCG@K			MPR
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20	
AI2V-Vanilla	0.039	0.076	0.132	0.019	0.024	0.028	0.024	0.036	0.050	0.136
AI2V-Pos	0.081	0.133	0.216	0.041	0.047	0.053	0.051	0.067	0.088	**0.125**
AI2V++	**0.083**	**0.138**	0.216	**0.045**	**0.052**	**0.057**	**0.054**	**0.072**	**0.091**	0.127
AI2V++ dot	0.078	0.136	**0.220**	0.040	0.048	0.053	0.049	0.068	0.089	0.124
I2V	0.028	0.052	0.097	0.013	0.016	0.019	0.017	0.025	0.036	0.162
POP	0.017	0.046	0.083	0.009	0.012	0.015	0.011	0.020	0.029	0.239
NCF	0.030	0.055	0.099	0.014	0.017	0.020	0.018	0.026	0.037	0.173
SASREC	0.056	0.109	0.185	0.025	0.030	0.035	0.031	0.048	0.067	0.129
LightGCN	0.025	0.053	0.096	0.013	0.016	0.019	0.016	0.024	0.036	0.240
NAIS	0.038	0.067	0.116	0.018	0.022	0.026	0.023	0.033	0.045	0.199

TABLE 4. Evaluation results on the Netflix dataset.

Model	HR@K			MRR@K			NDCG@K			MPR
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20	
AI2V-Vanilla	0.118	0.182	0.266	0.063	0.071	0.077	0.077	0.097	0.118	0.115
AI2V-Pos	0.158	0.227	0.319	0.094	0.103	0.109	0.110	0.132	0.155	**0.093**
AI2V++	**0.168**	**0.236**	**0.328**	**0.104**	**0.114**	**0.120**	**0.120**	**0.142**	**0.165**	0.095
AI2V++ dot	0.141	0.205	0.292	0.084	0.093	0.099	0.098	0.119	0.141	0.105
I2V	0.136	0.165	0.236	0.088	0.092	0.096	0.100	0.109	0.127	0.125
POP	0.092	0.107	0.122	0.062	0.069	0.074	0.065	0.069	0.072	0.360
NCF	0.117	0.130	0.156	0.074	0.075	0.077	0.084	0.089	0.095	0.216
SASREC	0.108	0.121	0.141	0.070	0.078	0.084	0.081	0.087	0.096	0.267
LightGCN	0.104	0.123	0.147	0.072	0.074	0.076	0.080	0.086	0.092	0.232
NAIS	0.097	0.132	0.201	0.071	0.075	0.080	0.077	0.088	0.106	0.162

TABLE 5. Evaluation results on the Yahoo! Music data.

Model	HR@K			MRR@K			NDCG@K			MPR
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20	
AI2V-Vanilla	0.116	0.186	0.274	0.059	0.068	0.074	0.073	0.095	0.118	0.043
AI2V-Pos	0.200	0.298	0.395	0.110	0.123	0.130	0.132	0.164	0.189	**0.035**
AI2V++	**0.240**	**0.330**	0.419	**0.134**	**0.146**	**0.152**	**0.160**	**0.189**	**0.212**	0.043
AI2V++ dot	0.231	0.329	**0.426**	0.121	0.134	0.141	0.148	0.180	0.204	0.036
I2V	0.072	0.122	0.199	0.037	0.044	0.049	0.045	0.062	0.081	0.053
POP	0.016	0.037	0.058	0.006	0.009	0.010	0.008	0.015	0.020	0.138
NCF	0.073	0.121	0.219	0.034	0.036	0.039	0.043	0.052	0.071	0.071
SASREC	0.080	0.141	0.226	0.036	0.044	0.05	0.047	0.066	0.088	0.043
LightGCN	0.082	0.149	0.241	0.036	0.047	0.053	0.051	0.069	0.090	0.075
NAIS	0.095	0.153	0.229	0.047	0.055	0.060	0.059	0.078	0.097	0.079

TABLE 6. Evaluation results on the Amazon Books dataset.

Model	HR@K			MRR@K			NDCG@K			MPR
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20	
AI2V-Vanilla	0.129	0.190	0.274	0.070	0.078	0.084	0.048	0.104	0.125	0.102
AI2V-Pos	0.151	0.214	0.297	0.088	0.097	0.102	0.104	0.124	0.145	**0.101**
AI2V++	0.161	0.222	0.302	0.099	0.107	0.113	0.115	0.134	0.154	0.108
AI2V++ dot	**0.172**	**0.235**	**0.312**	**0.108**	**0.117**	**0.122**	**0.124**	**0.144**	**0.164**	0.102
I2V	0.078	0.132	0.207	0.038	0.045	0.050	0.048	0.065	0.084	0.114
POP	0.005	0.019	0.030	0.003	0.005	0.005	0.003	0.008	0.011	0.416
NCF	0.028	0.051	0.089	0.013	0.016	0.018	0.016	0.024	0.033	0.196
SASREC	0.081	0.142	0.084	0.040	0.048	0.053	0.051	0.068	0.087	0.121
LightGCN	0.069	0.108	0.159	0.038	0.046	0.053	0.046	0.058	0.071	0.138
NAIS	0.126	0.181	0.243	0.069	0.076	0.080	0.083	0.100	0.116	0.134

In the second example, the target movie is "The Green Mile (1999)" and the model identified the movie "'Sixth Sense (1999)" as the user's highest-scored historical item. Both movies share similarities in their genre as drama films with a supernatural element, and also relate to the themes of crime and death. Additionally, the movie "Silence of the

TABLE 7. Evaluation results on the Moviesdat dataset.

Model	HR@K			MRR@K			NDCG@K			MPR
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20	
AI2V-Vanilla	0.036	0.059	0.095	0.017	0.020	0.023	0.022	0.029	0.038	0.146
AI2V-Pos	0.051	0.087	0.137	0.025	0.030	0.034	0.032	0.043	0.056	0.123
AI2V++	**0.061**	**0.102**	**0.164**	**0.031**	**0.036**	**0.040**	**0.038**	**0.051**	**0.067**	**0.118**
AI2V ++ dot	0.051	0.088	0.140	0.025	0.030	0.034	0.032	0.043	0.056	0.127
I2V	0.029	0.041	0.070	0.014	0.017	0.019	0.016	0.020	0.025	0.189
POP	0.004	0.009	0.020	0.002	0.002	0.003	0.002	0.004	0.007	0.332
NCF	0.028	0.039	0.068	0.013	0.015	0.017	0.014	0.018	0.023	0.208
SASREC	0.036	0.056	0.088	0.012	0.018	0.021	0.017	0.024	0.035	0.149
LightGCN	0.032	0.045	0.074	0.015	0.017	0.021	0.019	0.023	0.029	0.182
NAIS	0.005	0.010	0.019	0.003	0.003	0.004	0.003	0.005	0.007	0.109

Lambs (1991)'' was also identified as having a high score in the user's historical items. This movie is another crime drama that explores the complex relationship between a law enforcer and a criminal.

In the final example, the target item is ''The Dark Crystal (1982)''. The most similar item in the user's history, as determined by AI2V++, is the movie ''Labyrinth (1986)''. Both movies are fantasy/fictional movies directed and written by Jim Henson.

Next, we examine how the attention scores of users change in response to different items to be scored. The attention scores of a particular user are depicted in Fig. 7, where the left-side image represents the scores in the context of the classic romantic comedy ''Singing in the Rain (1952)''. In this case, the AI2V++ model gives more attention to another classic romantic comedy from the same era, ''Some Like it Hot (1959)'', followed by the romantic musical ''King and I (1956)''. The user's other items are science fiction films from the 90s and receive lower scores accordingly. While the model does not disregard them entirely, they are de-emphasized and given a lesser role. In contrast, the right-side image in Fig. 7 shows the attentive importance scores of the same user in the presence of the science fiction film ''Cube (1997)''. In this case, the model correctly identifies ''The Matrix (1999)'' and ''Star Trek: Insurrection (1998)'' as the most relevant, and the user's representation would adapt accordingly based on these scores.

Figure 8 presents another example based on a different user. The left-side image presents the importance scores of the model with respect to the romantic comedy ''Four Weddings and A Funeral (1994)'' starring Hugh Grant (an actor). We see that in the context of this target item, the model gives more attention to ''Notting Hill (1999)'' and ''When Harry Met Sally (1989)'', both are romantic comedies and the former also features Hugh Grant in the lead role. The right-side image in Fig. 8 presents a different target item for the same user. Now, the target item is the action movie ''Die Hard (1988)''. As can be seen, in the presence of this target item, the romantic comedies from the previous example are de-emphasized, and instead, two other movies come to the front: ''Indiana Jones and the Last Crusade (1989)'' and ''The Terminator (1984)''. Both are action-adventure movies from the '80s.

The above examples demonstrate that the attention weights of the different historic items enable model interpretability in AI2V++. This process can also be employed to ''explain'' the model's recommendations based on the user's historical items e.g., ''We recommend you 'X' because you watched 'Y'''.

V. DISCUSSION

In this paper, we have introduced AI2V++, an improved and extended version of our previous work on AI2V presented at a conference [19]. Unlike most CF algorithms that represent users as static vectors, AI2V++ models users dynamically based on the item being recommended. This approach mimics how the human brain operates when making decisions on different items, where different memories are activated and brought to the forefront. To achieve this, AI2V++ utilizes a neural cross-attention mechanism on the user's past items, where the target item is used as a query. As a result, the items in the user's history receive dynamic attention scores based on their relevance to the item being scored.

AI2V++ includes several algorithmic improvements over the conference presentation of AI2V [19]: (1) the integration of ordinal information into the context-target attention mechanism through a hierarchy of global and personal ordinal biases, and (2) the replacement of the categorical cross-entropy loss function used in AI2V with a binary cross-entropy loss function more suitable for multi-label classification problems. The effectiveness of these modifications is demonstrated through extensive quantitative and qualitative evaluations on five datasets, which show that the AI2V++ model outperforms several state-of-the-art recommender systems. Additionally, through attentive score analysis, we demonstrate interpretability with AI2V++ which can be harnessed for generating end-user explanations. To ensure reproducibility, the open-source code for AI2V++ is available on GitHub.

A. SPACE AND TIME COMPLEXITIES

In what follows, we wish to discuss the space and time complexities of the AI2V++ model in the context of real-world settings and in comparison to alternative algorithms.

The space complexity of the model is in line with that of classical MF models e.g. [11], [74]. In classical MF models,

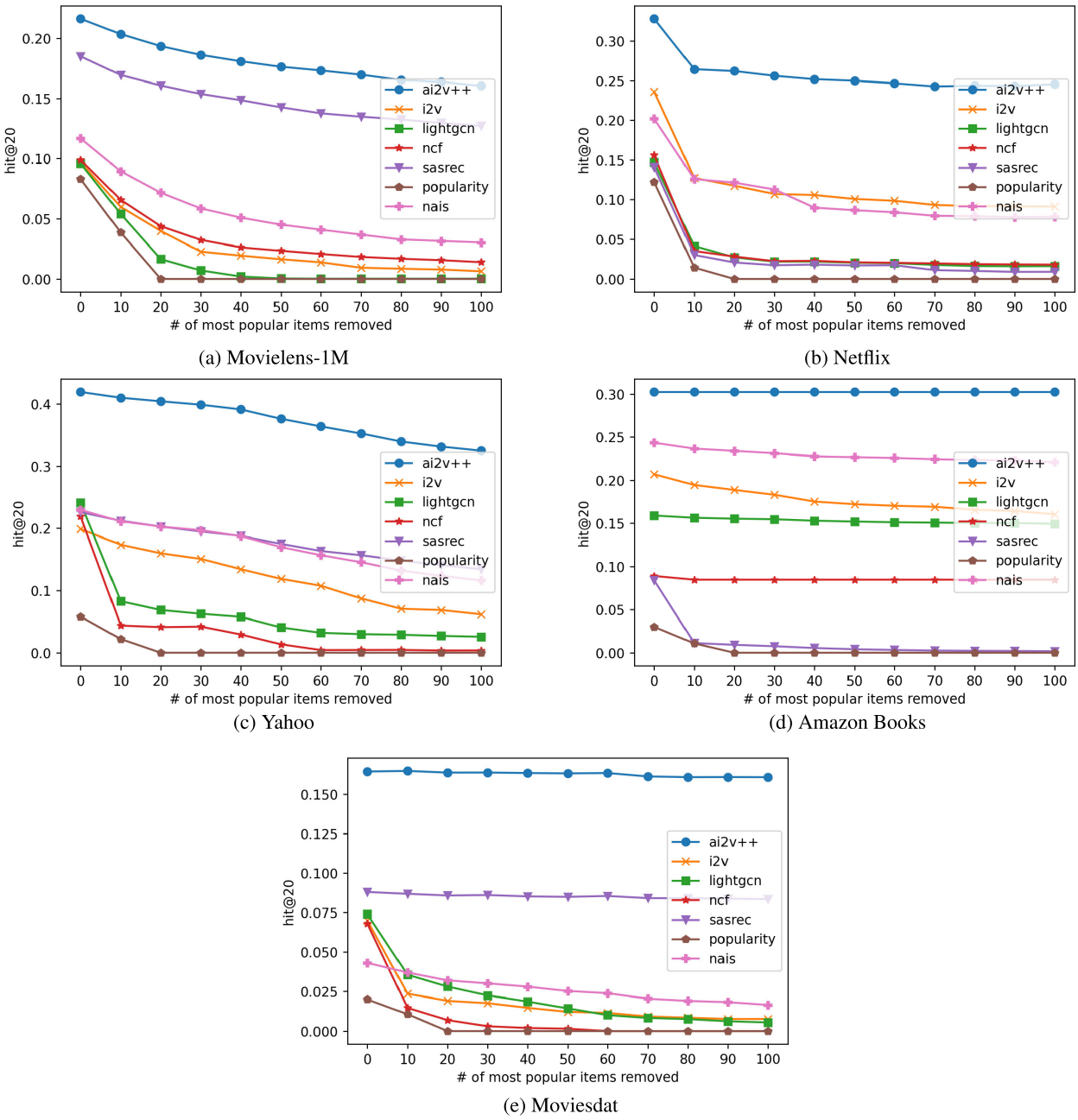


FIGURE 4. Accuracy of less popular items: This analysis measures the HR@20 when removing the most popular items from the test set. Initially (on the left), the 10 most popular items are removed. Then, at each point along the x-axis, the next 10 most popular items are also removed until all the top 100 popular items are removed. This analysis reveals how the different models cope when canceling out the most popular items that skew the dataset.

each user and each item is represented by a d-dimensional vector, hence the space complexity sums up to $\mathcal{O}((X+M)d)$, where X is the number of users, M is the number of items, and d is the dimensionality of the representations.

AI2V++ represents items similar to classic MF models, but different from these models, AI2V++ does not have explicit user representations. Instead, AI2V++ dynamically composes the user representation based on her items using a per-user attention mechanism. The attention parameters of AI2V++ consist of the projection matrixes $\mathbf{A}_c^i, \mathbf{A}_t^i \in R^{d \times d_\alpha}$, and $\mathbf{B}_c^i, \mathbf{B}_t \in R^{d \times d}$, where d is the dimensionality of the item embeddings, and d_α is the dimensionality of the attention space. We can easily assume that $d_\alpha \leq d$, since there is no benefit in inflating the representations. The projection matrixes $\mathbf{A}_c^i, \mathbf{A}_t^i$, and $\mathbf{B}_c^i$ are multiplied by the number of attention heads N. In addition, there is C_{max} ordinal biases per user and an additional set of C_{max} global biases. Putting it all together, the space complexity of AI2V++ is, therefore,

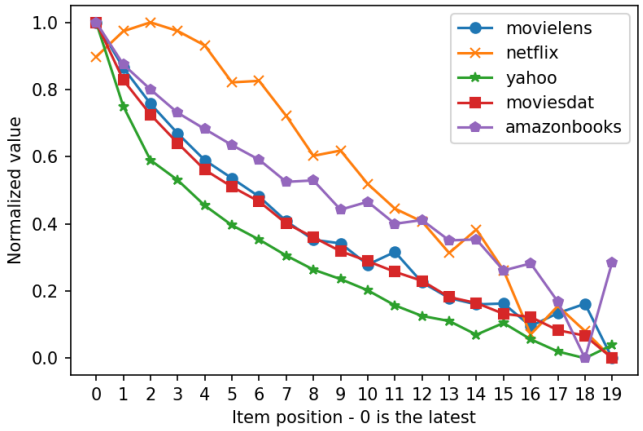


FIGURE 5. Ordinal bias weights of the last twenty items in the user's sequence. The items are ordered according to their consumption order. Each element in this vector corresponds to a certain position of an item in the user's sequence. In order to enable the comparison between different datasets on a single plot, we normalized the bias values of each model by the maximal bias value in that model.

$\mathcal{O}(Md + X \times (Nd^2 + C_{max}))$. While AI2V++'s space complexity is somewhat higher than that of a classic MF model, the difference is arguably modest and remains linear in the number of users and items.

In terms of training time complexity, since it is based on SGD its time complexity is linear with the number of parameters $\mathcal{O}(Md + X \times (Nd^2 + C_{max}))$ times the number of epochs and the number of samples in each epoch. The number of epochs and the number of samples varies from one dataset to another, but in general, the training time of AI2V++ is relatively fast. For example, we trained our model on a single NVIDIA V100 GPU and it took us 4 hours to train a model for Moviesdat dataset [60] which is the largest dataset we have used. On other datasets, the training process converged even faster. Arguably, the training time of AI2V++ is relatively modest in the realm of deep learning. Furthermore, our research code hasn't been optimized for production settings and can surely be improved by professional developers in order to reduce training time in industrial settings. Finally, training the algorithm is performed offline. Hence, its training time is of less importance with respect to its inference time.

While training a CF model is performed offline, the inference often needs to be performed online. Moreover, in order to pick the top recommendations, per-user ranking needs to be performed which incurs additional costs. Therefore, in CF, a model's inference time is usually of higher importance than its online training time.

As mentioned earlier, in basic CF algorithms such as MF [74], users are represented using fixed pre-computed latent vectors. At inference time, given a user and an item, the predicted affinity of the user to the items is simply given by computing the inner product between the user representation and the item representation. Hence, the time complexity for scoring a user-item pair is simply $\mathcal{O}(d)$. In contrast, the dynamic nature of AI2V++'s user representations inherently requires some additional computations. Given a user and an item, the model needs to compute the attention scores for the user's historical items with respect to the item first. Only then, it is possible to assemble the user vector which is used to score the item. Hence, if a user has k items in her history, the time complexity of scoring a single interaction is given by $\mathcal{O}(N \times k \times d)$. As can be seen, when compared to classical CF algorithms, two additional factors are added to AI2V++'s time complexity at inference: the first regards the number of historical items that the user interacted with i.e. k, and the second is a hyperparameter that determines the granularity of the multi-attentive user representation i.e. N. Let us briefly address both.

Empirically, when optimizing for the hyperparameter N, we found that it is best to set either $N = 1$ (MovieLens-1M, Yahoo! Music) or $N = 2$ (Netflix, Moviesdat). Hence, in practice, it can be argued that this number is very small and can be considered a constant. Note that even when $N = 1$, AI2V++ still enjoys significant advantages over classical CF algorithms.

The number of historical items per user k can be bounded by the total number of items in the catalog $k \leq M$. Hence, this can be a theoretical upper limit on the time complexity for each user-item prediction. However, for the lion's share of users, this upper limit is extremely inflated. CF datasets are known to be very sparse. That is due to the fact that most users only interact with a very small subset of the items in the catalog. For example, the sparsity level of the Netflix dataset is 98.82% [10] and the sparsity level in the Yahoo! Music dataset is 99.96% [12]. Moreover, such datasets exhibit a power-law distribution in which a long tail of users rate only a small number of items and only very few "heavy" users rate a large number of items. As a result, in the vast majority of cases $k \ll K$, and for those users, k can be considered as a constant number rather than a factor. For the few "heavy" users it is possible to employ different heuristics such as ignoring some of their historical items or pre-computing and caching their recommendations.

B. LIMITATIONS AND FUTURE WORK

The AI2V++ model presents a novel user representation scheme and demonstrates both state-of-the-art accuracy results as well as interpretability properties that throw light on the inner workings of the model. However, these advantages do not come without a cost. Arguably, the main limitation of the model is its inference time which stems from its dynamic user representation and the utilization of a neural cross-attention mechanism on users' historical items. In general, AI2V++'s inference time is several times higher than that of its classical predecessors. However, for the vast majority of users, if we treat both N and k as constants rather than factors, AI2V++'s inference time becomes $\mathcal{O}(d)$, similar to classical MF algorithms. Moreover, when compared to state-of-the-art algorithms based on neural attention such as in computer vision e.g., [39], [75] or in natural language

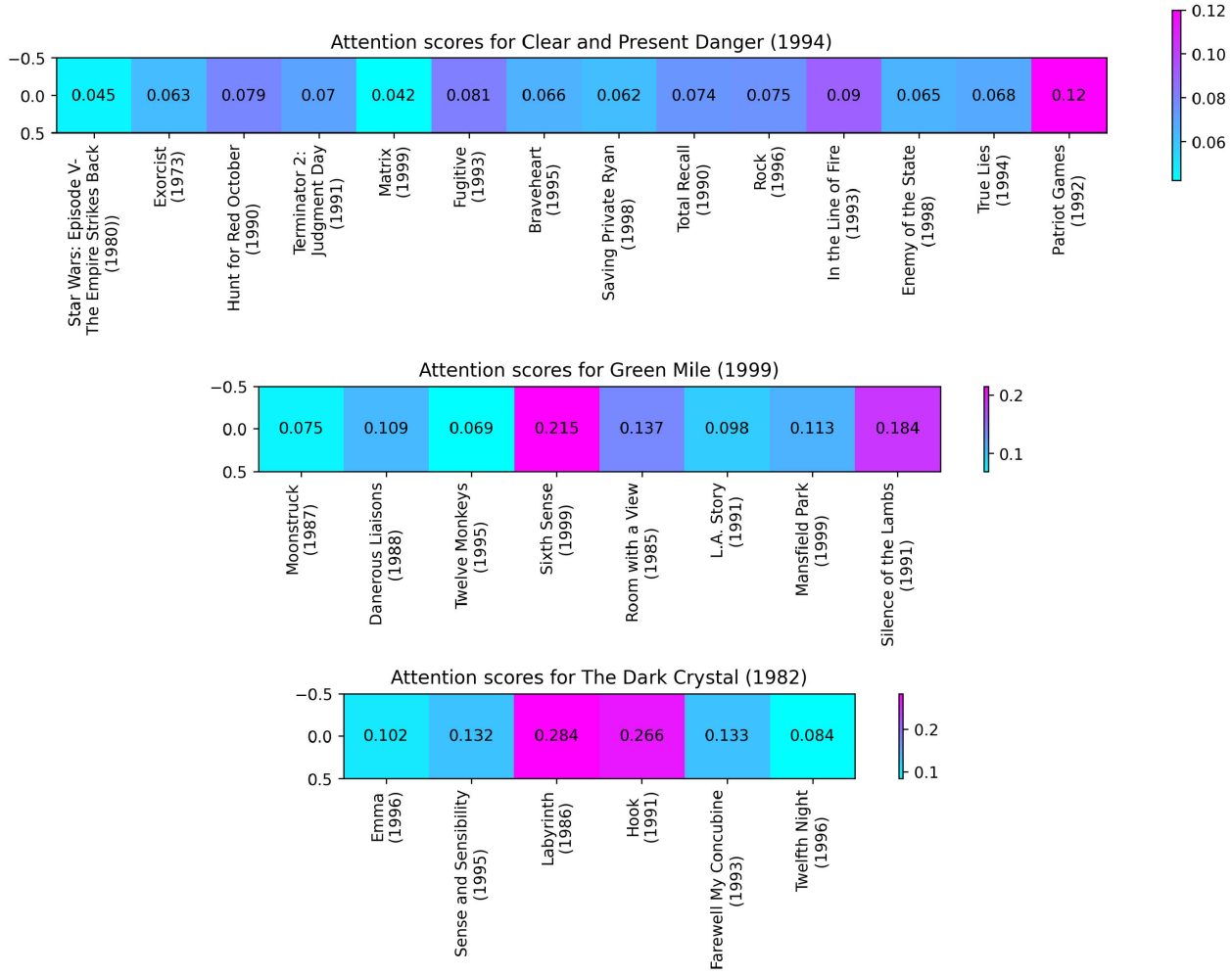


FIGURE 6. Attention scores for different users in the Movielens-1M dataset. Each plot represents a different user with her historical items (train items), chronologically ordered from left to right. The movie in the title is the target item. The figure presents the attention scores of the users' historical items with respect to the target item.

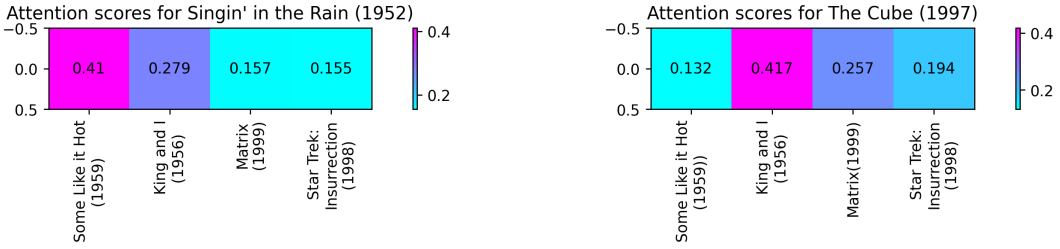


FIGURE 7. The change of importance (attention) scores for the same user in the presence of different target items: The left image depicts the importance scores in the presence of the classic romantic comedy "Singing in the Rain (1952)", while the right image depicts the importance score in the presence of the science fiction film "Cube (1997)."

processing e.g., [17], [20], [76], AI2V++'s inference time remains moderate.

In future work, we plan to investigate a distillation approach that treats AI2V++ as a "teacher" model and trains a nimble "student" model which learns to reconstruct AI2V++'s cross-attention mechanism using regression. A similar approach is already been used successfully in the field of natural language processing [57], where it has been applied in order to mitigate the expensive cross-attention operation at the inference phase of models such as BERT [20] and XLNet [77].

The explainability of AI2V++ represents another area for future research. Interpretability and explainability are two related terms that are often used interchangeably [78],

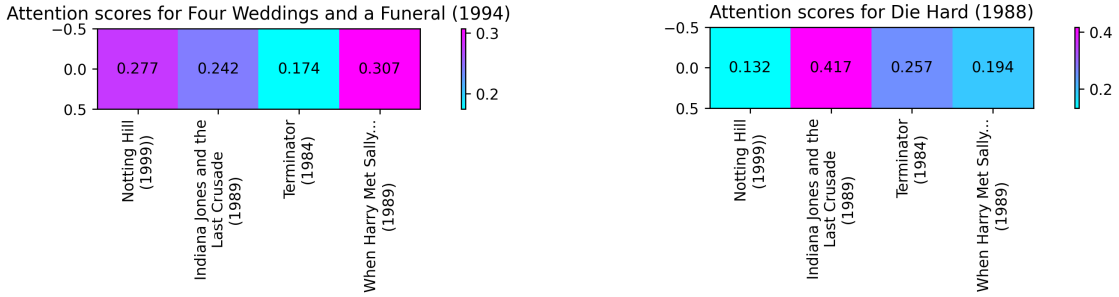


FIGURE 8. The change of importance (attention) scores for the same user in the presence of different target items: The left image depicts the importance scores in the presence of the romantic comedy "Four Weddings and a Funeral (1994)", while the right image depicts the importance score in the presence of the action film "Die Hard (1988)."

[79]. Interpretability refers to the ability to comprehend the inner workings of the model by someone knowledgeable in the field, while explainability relates to the ability to provide clear explanations to end-users that rationalize the recommendations. In this study, we demonstrated effective interpretability properties that can be leveraged for intuitive explanations. However, evaluating explanations in recommender systems is a complex task that varies according to the explanations' aim, such as transparency, scrutability, trust, effectiveness, persuasiveness, efficiency, and satisfaction. Moreover, the usefulness of an explanation depends on objective and subjective system aspects, user experience, situational, interaction, and personal characteristics. Although we cannot evaluate these aspects in the scope of the current paper, we hope that the utility of the proposed approach is evident. Future studies can investigate how to improve the explainability of AI2V++ further.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," in *Recommender Systems Handbook*. 2021, pp. 1–35.

[2] G. M. Davies and D. M. Thomson, *Memory in Context: Context in Memory*. Hoboken, NJ, USA: Wiley, 1988.

[3] A. K. Anderson, Y. Yamaguchi, W. Grabski, and D. Lacka, "Emotional memories are not all created equal: Evidence for selective memory enhancement," *Learn. Memory*, vol. 13, no. 6, pp. 711–718, Nov. 2006.

[4] S. M. Smith and E. Vela, "Environmental context-dependent memory: A review and meta-analysis," *Psychonomic Bull. Rev.*, vol. 8, no. 2, pp. 203–220, Jun. 2001.

[5] S. Sukhbaatar, D. Ju, S. Poff, S. Roller, A. Szlam, J. Weston, and A. Fan, "Not all memories are created equal: Learning to forget by expiring," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9902–9912.

[6] L. Chen, M. De Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro, "Human decision making and recommender systems," *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 3, pp. 1–7, 2013.

[7] E. Lex and M. Schedl, "Psychology-informed recommender systems tutorial," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 714–717.

[8] J. R. Bettman, M. F. Luce, and J. W. Payne, "Constructive consumer choice processes," *J. Consum. Res.*, vol. 25, no. 3, pp. 187–217, Dec. 1998.

[9] S. Lichtenstein and P. Slovic, *The Construction of Preference*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[10] J. Bennett and S. Lanning, "The Netflix prize," in *Proc. KDD Cup Workshop*, New York, NY, USA, 2007, p. 35.

[11] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 263–272.

[12] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The Yahoo! Music dataset and KDD-cup'11," in *Proc. KDD Cup*, 2012, pp. 3–18.

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

[14] N. Koenigstein and Y. Koren, "Towards scalable and accurate item-oriented recommendations," in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 419–422.

[15] O. Barkan and N. Koenigstein, "ITEM2VEC: Neural item embedding for collaborative filtering," in *Proc. IEEE 26th Int. Workshop Mach. Learn. for Signal Process. (MLSP)*, Sep. 2016, pp. 1–6.

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[18] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020, *arXiv:2001.04451*.

[19] O. Barkan, A. Caciularu, O. Katz, and N. Koenigstein, "Attentive item2vec: Neural attentive user representations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3377–3381.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[21] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond, and J. Basilico, "Deep learning for recommender systems: A Netflix case study," *AI Mag.*, vol. 42, no. 3, pp. 7–18, Nov. 2021.

[22] B. Selma, B. Narhimene, and R. Nachida, "Deep learning for recommender systems: Literature review and perspectives," in *Proc. Int. Conf. Recent Adv. Math. Informat. (ICRAMI)*, Sep. 2021, pp. 1–7.

[23] Y. Wei, M. Langer, F. Yu, M. Lee, J. Liu, J. Shi, and Z. Wang, "A GPU-specialized inference parameter server for large-scale deep recommendation models," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 408–419.

[24] H. Chen, Y. Lin, M. Pan, L. Wang, C.-C.-M. Yeh, X. Li, Y. Zheng, F. Wang, and H. Yang, "Denoising self-attentive sequential recommendation," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 92–101.

[25] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 111–112.

[26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. World Wide Web Conf. Steering Committee*, 2017, pp. 173–182.

[27] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *Proc. 14th ACM Conf. Recommender Syst.*, Sep. 2020, pp. 240–248.

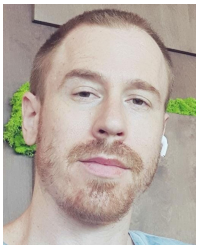
[28] S. Rendle, W. Krichene, L. Zhang, and Y. Koren, "Revisiting the performance of iALS on item recommendation benchmarks," in *Proc. 16th ACM Conf. Recommender Syst.*, Sep. 2022, pp. 427–435.

[29] 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. Retr.*, 2019, pp. 165–174.

[30] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[32] T. Ebesu, B. Shen, and Y. Fang, "Collaborative memory network for recommendation systems," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 515–524.
[33] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai, "HOP-Rec: high-order proximity for implicit recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 140–144.
[34] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 974–983.
[35] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
[36] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2020, pp. 639–648.
[37] A. D. S. Correia and E. L. Colombini, "Attention, please! A survey of neural attention models in deep learning," 2021, *arXiv:2103.16775*.
[38] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, pp. 1–41, Sep. 2021.
[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[41] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. ICML*, 2021, pp. 8748–8763.
[42] X. Zhou and Y. Li, "Large-scale modeling of mobile user click behaviors using deep learning," in *Proc. 15th ACM Conf. Recommender Syst.*, 2021, pp. 473–483.
[43] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206.
[44] 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.
[45] R. He, W.-C. Kang, and J. McAuley, "Translation-based recommendation," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 161–169.
[46] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.
[47] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-K gains for session-based recommendations," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 843–852.
[48] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 565–573.
[49] 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 *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1441–1450.
[50] W. L. Taylor, "'Cloze procedure': A new tool for measuring readability," *Journalism Quart.*, vol. 30, no. 4, pp. 415–433, 1953.
[51] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018.
[52] N. Tintarev and J. Masthoff, "Beyond explaining single item recommendations," in *Recommender Systems Handbook*. 2022, pp. 711–756.
[53] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, "Explainable AI methods—A brief overview," in *Proc. Int. Workshop Extending Explainable AI Beyond Deep Models Classifiers*. Cham, Switzerland: Springer, 2022, pp. 13–38.
[54] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in *Proc. CCF Int. Conf. Natural Language Process. Chinese Comput.* Cham, Switzerland: Springer, 2019, pp. 563–574.
[55] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a 'right to explanation,'" *AI Mag.*, vol. 38, no. 3, pp. 50–57, 2017.
[56] K. Swearingen and R. Sinha, "Beyond algorithms: An HCI perspective on recommender systems," in *Proc. ACM SIGIR Workshop Rec. Syst.*, vol. 13, 2001, pp. 1–11.
[57] O. Barkan, N. Razin, I. Malkiel, O. Katz, A. Caciularu, and N. Koenigstein, "Scalable attentive sentence pair modeling via distilled sentence embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3235–3242.
[58] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Dec. 2015.
[59] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2011, pp. 1–35.
[60] *The Movies Dataset*. [Online]. Available: https://www.kaggle.com/rounakbanik/the-movies dataset
[61] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 43–52.
[62] M. Chen, "Performance evaluation of recommender systems," *Int. J. Performability Eng.*, vol. 13, no. 8, p. 1246, 2017.
[63] A. P. Singh and G. J. Gordon, "A unified view of matrix factorization models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2008, pp. 358–373.
[64] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Apr. 2001, pp. 285–295.
[65] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 549–558.
[66] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf.*, 2018, pp. 689–698.
[67] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, 2015.
[68] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.
[69] A. Lydia and S. Francis, "Adagrad—An optimizer for stochastic gradient descent," *Int. J. Inf. Comput. Sci.*, vol. 6, no. 5, pp. 566–568, 2019.
[70] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 39–46.
[71] G. Dror, N. Koenigstein, and Y. Koren, "Web-scale media recommendation systems," *Proc. IEEE*, vol. 100, no. 9, pp. 2722–2736, Sep. 2012.
[72] O. Celma, "The long tail in recommender systems," in *Music Recommendation and Discovery*. Berlin, Germany: Springer, 2010, pp. 87–107.
[73] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 447–456.
[74] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[76] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," in *Proc. Adv. Neur. Inf. Process. Sys.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Curran Associates, 2020, pp. 1877–1901.
[77] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[78] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Mar. 2018, pp. 80–89.
[79] H. J. Escalante, S. Escalera, I. Guyon, X. Baro, Y. Gucluturk, U. Guclu, M. van Gerven, and R. van Lier, *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Berlin, Germany: Springer, 2018.

KEREN GAIGER received the B.Sc. degree in information systems from the Technion—Israel Institute of Technology, Haifa, Israel, in 2017. She is currently pursuing the M.Sc. degree with Tel Aviv University, Tel Aviv, Israel. From 2017 to 2020, she was a Data Scientist with two Israeli startups, first with AdTech company and a Security company. She is a Researcher with Lightricks, a company that develops video and image editing mobile apps.



OREN BARKAN received the B.Sc. and M.Sc. degrees (cum laude) in computer science from Hebrew University, and the Ph.D. degree from the School of Computer Science, Tel Aviv University, Israel. He is currently an Assistant Professor of computer science with The Open University of Israel.

SHIR TSIPORY-SAMUEL received the B.Sc. degree in industrial engineering and management from Tel Aviv University, Tel Aviv, Israel, in 2018, where she is currently pursuing the M.Sc. degree in industrial engineering and management. In 2017, she joined JFrog as a Security Data Analyst, where she researched vulnerabilities and maintained JXray's database. In 2021, she joined Bionic as a Research Analyst, researching code as part of the static analysis team and building better analysis visualizations.

NOAM KOENIGSTEIN received the B.Sc. degree (cum laude) in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2007, the M.Sc. degree in electrical engineering from Tel Aviv University, Tel Aviv, Israel, in 2009, and the Ph.D. degree from the School of Electrical Engineering, Tel Aviv University. In 2011, he joined the Xbox Machine Learning Research Team, Microsoft, where he developed the algorithm for Xbox recommendations serving millions of users worldwide. Later, he managed the Recommendations Research Team, Microsoft Store. In 2017, he joined Citi Bank's Israeli Innovation Laboratory as the Senior VP Head of data science overseeing all data science activities with the Israeli Research Center. In 2018, he joined the Industrial Engineering Department, Tel Aviv University, as a Senior Lecturer (Associate Professor). He currently heads the Applied Machine Learning Laboratory (AML Lab), where his students are working on applying machine learning algorithms to a diverse set of real-world problems.