

# Attention Mixtures for Time-Aware Sequential Recommendation

Viet-Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, Romain Hennequin  
Deezer Research, Paris, France  
research@deezer.com

## ABSTRACT

Transformers emerged as powerful methods for sequential recommendation. However, existing architectures often **overlook the complex dependencies between user preferences and the temporal context**. In this short paper, we introduce MOJITO, an improved Transformer sequential recommender system that addresses this limitation. MOJITO leverages Gaussian mixtures of attention-based temporal context and item embedding representations for sequential modeling. Such an approach permits to accurately predict which items should be recommended next to users depending on past actions and the temporal context. We demonstrate the relevance of our approach, by empirically outperforming existing Transformers for sequential recommendation on several real-world datasets.

## CCS CONCEPTS

• **Information systems** → *Recommender systems; Personalization.*

## KEYWORDS

Sequential Recommendation, Attention Mixtures, Transformers.

## ACM Reference Format:

Viet-Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, Romain Hennequin. 2023. Attention Mixtures for Time-Aware Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*, July 23–27, 2023, Taipei, Taiwan. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3539618.3591951>

## 1 INTRODUCTION

Sequential recommendation (SR) is essential to online platforms such as movie and music streaming services [4, 6, 17, 29, 34, 35]. By modeling temporal sequences of user actions [9, 29, 42], SR systems predict the best content to recommend at a given time based on past interactions, e.g., the best songs to extend a listening session [26]. Over the past years, researchers have devoted significant efforts to improving SR [9, 13, 14, 20, 31, 39, 52, 56]. In particular, Transformers [40], the most recent class of neural architectures for sequence modeling, emerged as powerful SR methods [8, 15, 17, 35, 43, 47, 53].

However, most existing Transformer systems for SR omit or insufficiently exploit the *temporal context* associated with user actions [7, 11, 20, 46, 51]. While they process the relative position of an interaction within a sequence, these systems often overlook

time-related information such as the hour or the day of each interaction. Yet, such factors can be crucial in real-world SR problems. For instance, in the music domain, Hansen et al. [11] observed that capturing temporal elements such as the hour of the day helps predict the following songs users listen to on a music streaming service. In another study focusing on various domains, from music to e-commerce, Ye et al. [51] identified several time-dependent patterns in interaction sequences. This includes seasonal behaviors, e.g., Christmas songs being more consumed in December, and periodic behaviors, e.g., buying a new toothbrush every three months.

This paper introduces Mixture Of Joint Item and Temporal cOntext (MOJITO), an improved Transformer SR system overcoming this limitation. MOJITO relies on the assumption that users tend to consume similar items, content-wise, for a given temporal context. Specifically, it jointly considers user-item interactions and the temporal context to learn two different attention matrices subsequently combined within Gaussian mixtures of temporal context and item embedding representations. This allows for an effective time-aware sequential modeling of user actions. We provide comprehensive experiments on various real-world datasets covering application domains ranging from movie to music recommendation. Our results emphasize that MOJITO empirically outperforms existing Transformers for SR, including models previously proposed to capture the dependencies between user preferences and the temporal context [7, 20, 30, 46]. Consequently, we believe MOJITO will be helpful for researchers and practitioners eager to thoroughly leverage time-related contextual aspects when tackling SR problems. We publicly release our code on GitHub to facilitate the usage of our system.

This paper is organized as follows. In Section 2, we formally present the SR problem and review related work. In Section 3, we introduce our proposed system for time-aware SR. We discuss our experimental analysis in Section 4, and conclude in Section 5.

## 2 PRELIMINARIES

### 2.1 Problem Formulation

We consider a set  $\mathcal{U}$  of users and a set  $\mathcal{V}$  of items on an online service. For each  $u \in \mathcal{U}$ , we observe  $n_u \geq 1$  sequences of interactions on the service, e.g., listening sessions on a music streaming service. We denote by  $S_{ui} = (v_1, v_2, \dots, v_L)$  the  $i^{\text{th}}$  interaction sequence of  $u$ , with  $i \in \{1, \dots, n_u\}$ ,  $v_t \in \mathcal{V}$ ,  $\forall t \in \{1, \dots, L\}$ , and a length<sup>1</sup>  $L \in \mathbb{N}^*$ .

Along with each  $S_{ui}$ , we observe a *contextual sequence*  $C_{ui} = (c_1, c_2, \dots, c_L)$ . Each element  $c_t$  (with  $t \in \{1, \dots, L\}$ ) is a tuple associating the  $t^{\text{th}}$  interaction of  $S_{ui}$  with  $C \in \mathbb{N}^*$  different types of *time-related contextual information*, such as the month (from January to December), the day of the month (from 1 to 31), or the day of the week (from Monday to Sunday) of this interaction. Formally, we set  $c_t = (c_{t1}, c_{t2}, \dots, c_{tC})$ , where  $c_{tj} \in C_j$ ,  $\forall j \in \{1, \dots, C\}$ , with

<sup>1</sup>Like Kang and McAuley [17], we fix the length  $L$  to simplify our presentation. Our work easily extends to sequences of varying lengths  $L_{ui} < L$ , by considering that the  $L - L_{ui}$  first elements of such sequences are null values, acting as “padding” items [17].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-9408-6/23/07...\$15.00  
<https://doi.org/10.1145/3539618.3591951>

$C_j$  denoting the set of possible values for the  $j^{\text{th}}$  context type. We order these values in each set  $C_j$  to account for temporal proximity, e.g., to capture that January is closer to February than to July.

In such a setting, SR consists in predicting the next item  $v_{L+1}$  a user will interact with, in the observed temporal context  $c_{L+1}$ .

## 2.2 Related Work

Addressing an SR problem usually requires to jointly account for:

- (1) “Long-term” preferences: this term designates intrinsic user preferences, independent of the current context [42].
- (2) “Short-term” preferences: on the contrary, this term refers to how the context affects preferences, including how recent interactions alter the perception of a recommendation [1, 9].

In recent years, several SR systems have been proposed to model these two aspects, often building upon advances in deep learning for sequence modeling [9, 10, 13, 14, 17, 20, 31, 39, 52, 53, 55, 56]. This paper focuses on Transformers [40] for SR. Using attention mechanisms [40] for sequence modeling, they have recently shown competitive results on SR problems [5, 8, 15, 17, 35, 43, 45, 47, 50, 53].

**2.2.1 Transformers for SR.** SASRec [17] was the first system to leverage self-attention to identify relevant items among temporal sequences for SR. BERT4Rec [35] subsequently implemented bidirectional self-attention techniques. To combine long-term and short-term preferences more carefully, AttRec [53] adopted a multi-task metric learning framework, while SSE-PT [43] concatenated user “embedding” vector representations with each item in the sequence. Lately, FISSA [15] incorporated an attentive version of the Factored Item Similarity Model (FISM) [16] for long-term user preference, and used an item similarity-based gating mechanism to merge short-term and long-term representations, with promising performances. Other recent work also combined attention mechanisms for SR with graph neural networks [28, 44] or models based on contrastive learning [8, 21], studied fairness-aware attention-based SR [19], and presented more effective training strategies [27].

**2.2.2 Transformers for Time-Aware SR.** Despite their success, these Transformers were also criticized for not fully exploiting *temporal* factors [7, 11, 20, 46, 51]. While their attention mechanisms exploit relative positions in interaction sequences for preference modeling, they neglect the temporal context of each interaction, captured by  $C_{ui}$  sequences in Section 2.1. Yet, as illustrated in the introduction, time-related factors such as the current month, day, or hour can significantly modify user needs and preferences [4, 11, 51].

Attempting to tackle this problem, TiSASRec [20] enhanced SASRec by analyzing time intervals between interactions using self-attention operations. TASER [51] combined pairwise relative and pointwise absolute time patterns in the same model. MEANTIME [7] leveraged multiple self-attention heads to learn several embedding representations, encoding different temporal patterns. TimelyRec [46] modeled heterogeneous temporal patterns in user behaviors, by processing hierarchical time granularities. Recently, CARCA [30] relied on contextual correlations computed via cross-attention to provide time-aware recommendations.

Nonetheless, we believe that time-aware SR remains incompletely addressed. While existing time-aware Transformers gather item and temporal representations using simple concatenations or

additions, such an approach oversimplifies the complex relationship between user preferences and the temporal context. Moreover, except MEANTIME [7], the aforementioned methods suffer from redundant attention heads problems [2, 23, 41]. The remainder of this paper will emphasize that overcoming these limitations permits to better model preferences and empirically improve time-aware SR.

## 3 A MIXTURE SYSTEM FOR TIME-AWARE SR

### 3.1 General Overview

This Section 3 introduces MOJITO. As illustrated in Figure 1, our system involves two components. The first one, described in Section 3.2, models short-term intents influenced by interactions and the temporal context. The second one, presented in Section 3.3, captures long-term preferences. Each component returns a *relevance score*, denoted  $r_{L+1}^{\text{short}}(v)$  and  $r_{L+1}^{\text{long}}(v)$  respectively. They estimate how likely each item  $v \in \mathcal{V}$  would fit as an extension of an observed interaction sequence. To select the items to recommend, we combine scores as follows, for some hyperparameter  $\lambda \in [0, 1]$ :

$$r_{L+1}(v) = \lambda r_{L+1}^{\text{short}}(v) + (1 - \lambda) r_{L+1}^{\text{long}}(v). \quad (1)$$

### 3.2 Short-Term Representation

We leverage self-attention mechanisms [17, 40] for time-aware short-term representation learning. As this approach is known to suffer from redundant heads<sup>2</sup> problems where different attention heads actually learn the same information [2, 23, 41], we build upon recent efforts from Nguyen et al. [24] and enable our attention heads to interact via a shared pool of global attention matrices. Nguyen et al. [24] sampled the local attention matrix of each head using an admixture model [3]. In our work, we enhance the diversity in the representation of global matrices by introducing different inductive semantic biases to guide their learning, as described below.

**3.2.1 Embedding Layer.** This first block processes the  $S_{ui}$  and  $C_{ui}$  sequences from Section 2.1. We denote by  $\mathbf{M}^I \in \mathbb{R}^{|\mathcal{V}| \times d}$  a learnable *item embedding matrix*, in which rows are  $d$ -dimensional embedding vectors representing each item, for some dimension  $d \in \mathbb{N}^*$ . Using such a notation, a sequence  $S_{ui}$  can be represented as an embedding matrix, as follows:  $\mathbf{E}_{S_{ui}}^I = [\mathbf{m}_{v_1}^I; \mathbf{m}_{v_2}^I; \dots; \mathbf{m}_{v_L}^I]^\top \in \mathbb{R}^{L \times d}$ .

We leverage the *translation-invariant time kernel* of Xu et al. [49] to learn  $d$ -dimensional *temporal context embedding vectors* compatible with self-attention operations. Specifically, each  $c_{ti} \in c_t$  is mapped to a vector  $\mathbf{m}_{c_{ti}}^C \in \mathbb{R}^d$  using a Mercer kernel function [48]. We concatenate and inject these vectors into a linear layer function  $g(\cdot)$  to obtain an embedding representation  $\mathbf{m}_{c_t}^C = g([\mathbf{m}_{c_{t1}}^C; \mathbf{m}_{c_{t2}}^C; \dots; \mathbf{m}_{c_{t|C_t|}}^C]) \in \mathbb{R}^d$  for each context tuple  $c_t$ . We obtain an *item/context embedding matrix* combining item and context embeddings:  $\mathbf{E}_{S_{ui}, C_{ui}} = [[\mathbf{m}_{v_1}^I; \mathbf{m}_{c_1}^C]; \dots; [\mathbf{m}_{v_L}^I; \mathbf{m}_{c_L}^C]]^\top \in \mathbb{R}^{L \times 2d}$ .

Lastly, to model the influence of positions in sequences, we enrich  $\mathbf{E}_{S_{ui}, C_{ui}}$  with learnable *position embeddings*  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_L]^\top \in \mathbb{R}^{L \times 2d}$ , obtaining our final *input matrix*  $\mathbf{X}^{(0)} = [\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_L^{(0)}]^\top \in \mathbb{R}^{L \times 2d}$ , where  $\mathbf{x}_l^{(0)} = [\mathbf{m}_{v_l}^I; \mathbf{m}_{c_l}^C] + \mathbf{p}_l, \forall l \in \{1, \dots, L\}$ . We aim to distinguish different influences of the same item in different positions.

<sup>2</sup>We refer, for instance, to the experiments of Kang and McAuley [17], showing that increasing the number of heads in SASRec does not improve sequential recommendation.

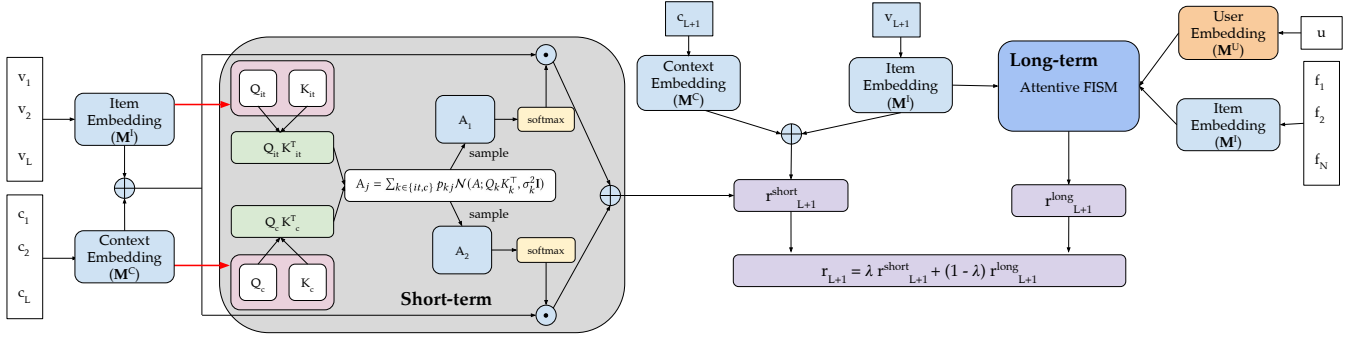


Figure 1: Architecture of MOJITO for time-aware SR using attention mixtures of temporal context and item embeddings.

**3.2.2 Attention Mixtures.** We pass  $\mathbf{X}^{(0)}$  through  $B \in \mathbb{N}^*$  stacked *self-attention blocks* (SAB) [40]. The output of the  $b^{\text{th}}$  block is  $\mathbf{X}^{(b)} = \text{SAB}^{(b)}(\mathbf{X}^{(b-1)})$ , for  $b \in \{1, \dots, B\}$ . The SAB contains a *self-attention layer*  $\text{SAL}(\cdot)$  with  $H \in \mathbb{N}^*$  heads mixing context and item embedding representations, followed by a *feed-forward layer*  $\text{FFL}(\cdot)$ :

$$\text{SAL}(\mathbf{X}) = \text{MultiHead}(\{\mathbf{X}_j^{\text{Att}}\}_{j=1}^H) = \text{Concat}(\mathbf{X}_1^{\text{Att}}, \dots, \mathbf{X}_H^{\text{Att}}) \mathbf{W}^O,$$

$$\text{SAB}(\mathbf{X}) = \text{FFL}(\text{SAL}(\mathbf{X})) = \text{ReLU}(\mathbf{X}^{\text{Att}} \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2, \quad (2)$$

where  $\mathbf{X} \in \mathbb{R}^{L \times 2d}$  is the input of each block,  $\mathbf{W}^O \in \mathbb{R}^{2Hd \times 2d}$  is the projection matrix for the output, and  $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{2d \times 2d}$  and  $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{1 \times 2d}$  are weights and biases for the two layers of the FFL network.  $\mathbf{X}_j^{\text{Att}} = \text{softmax}(\mathbf{A}_j / \sqrt{d}) \mathbf{V}$  is the output of each head. In this work, we sample  $\mathbf{A}_j$  from a *Gaussian mixture probabilistic model*, automatically learning the relative importance of the temporal context and previous item interactions for sequential modeling:

$$\mathbf{A}_j \sim \sum_{k \in \{it, c\}} p_{kj} \mathcal{N}(\mathbf{A}; \mathbf{Q}_k \mathbf{K}_k^T, \sigma_k^2 \mathbf{I}), \quad \sum_{k \in \{it, c\}} p_{kj} = 1, p_{kj} \geq 0, \quad (3)$$

where  $\mathbf{Q}_k = \mathbf{X}_k \mathbf{W}_Q^k$ ,  $\mathbf{K}_k = \mathbf{X}_k \mathbf{W}_K^k$  and  $\mathbf{V} = \mathbf{X} \mathbf{W}_V$ , with  $\mathbf{W}_Q^k, \mathbf{W}_K^k \in \mathbb{R}^{d \times d}$  and  $\mathbf{W}_V \in \mathbb{R}^{2d \times 2d}$  are the projected value matrices.

**3.2.3 Prediction.** We use a *latent factor model* [17, 20] to estimate the relevance of any  $v \in \mathcal{V}$  to extend  $S_{ui}$ , in the observed context  $\mathbf{c}_{L+1}$ . We obtain the *short-term relevance score* as follows:

$$r_{L+1}^{\text{short}}(v) = \mathbf{F}_L^{(B)} \mathbf{e}_v^T, \quad (4)$$

where  $\mathbf{F}_L^{(B)}$  is the model output at position  $L$  after  $B$  attention blocks and  $\mathbf{e}_v = [\mathbf{m}_v^I; \mathbf{m}_{\mathbf{c}_{L+1}}^C]^T \in \mathbb{R}^{2d}$ .

### 3.3 Long-Term Representation

We now turn to long-term representation learning. At this stage, we found it crucial to capture the *heterogeneity* of user preferences.

**3.3.1 Attentive FISM.** In real-world applications, users often have diverse preferences [25, 37, 54]. For instance, one can simultaneously like metal and classical music. Hence, the practice of representing long-term user tastes using a single embedding vector (as in AttRec [53] or SSE-PT [43]) has been shown to be limiting when preferences are dispersed in the embedding space [37, 38].

To overcome this issue, we use an attentive version of FISM [16] to represent users in MOJITO. This model, also incorporated in the

FISSA system for SR [15], allows user representations to flexibly *translate* depending on the current target item representation [38]. Contrary to FISSA, we learn user representations using  $N \in \mathbb{N}^*$  randomly selected items from the user’s past history, instead of using only items from the currently observed sequence. We aim to model more diverse long-term interests by adopting such a strategy.

Formally, denoting by  $\mathcal{F} = \{f_1, \dots, f_N\}$  the  $N$  selected items, the long-term preference of user  $u$  for an item  $v \in \mathcal{V}$  at any time  $l+1$  is estimated using a weighted aggregation of item representations:

$$\mathbf{y}_{L+1}^u(v) = \mathbf{m}_u + \sum_{f \in \mathcal{F} \setminus \{v\}} \frac{e^{\mathbf{m}_f^T \mathbf{m}_v}}{\sum_{f' \in \mathcal{F} \setminus \{v\}} e^{\mathbf{m}_{f'}^T \mathbf{m}_v}} \mathbf{m}_f. \quad (5)$$

**3.3.2 Prediction.** Similarly to Section 3.2.3, we obtain *long-term relevance scores* as follows:

$$r_{L+1}^{\text{long}}(v) = \mathbf{y}_{L+1}^u(v) \mathbf{m}_v^T. \quad (6)$$

We emphasize that this operation does *not* involve any temporal context information, as long-term preferences are assumed to be independent of the context by definition (see Section 2.2).

### 3.4 Training Procedure

MOJITO involves various weight matrices to optimize. For this purpose, we consider a training set  $\mathcal{S}$  of interaction and context sequences. For each interaction sequence  $S_{ui}$ , we create sub-sequences composed of the  $l$  first items of  $S_{ui}$  for  $l \in \{1, \dots, L\}$ . We denote by  $v_{S_{ui}, l+1}$  the  $(l+1)^{\text{th}}$  ground truth item extending each sequence. Additionally, we sample a “negative” item  $o_{S_{ui}, l+1} \in \mathcal{V} \setminus v_{S_{ui}, l+1}$  for each sub-sequence. We expect our system to return high relevance scores for  $v_{S_{ui}, l+1}$  items, and lower relevance scores for negative items. To achieve this, we optimize model weights via gradient descent minimization of the loss  $\mathcal{L} = \lambda \mathcal{L}^{\text{short}} + (1 - \lambda) \mathcal{L}^{\text{long}}$  (we recall that  $\lambda \in [0, 1]$ ), a linear combination of the  $\mathcal{L}^{\text{short}}$  and  $\mathcal{L}^{\text{long}}$  losses defined as follows:

$$\mathcal{L}^x = - \sum_{S \in \mathcal{S}} \sum_{l=1}^L [\log(\sigma(r_{l+1}^x(v_{S, l+1}))) + \log(1 - \sigma(r_{l+1}^x(o_{S, l+1})))] \quad (7)$$

for  $x \in \{\text{short}, \text{long}\}$  and the sigmoid function  $\sigma(a) = \frac{1}{1+e^{-a}}$ .

**Table 1: Sequential Recommendation (SR) on all datasets using MOJITO and other baselines. Scores are computed on test items. Models recommend ranked lists of 10 items. Bold numbers are the best scores. Underlined numbers are the second-best ones.**

Type	Model	MovieLens ( $k^{\text{user}} = 10, k^{\text{item}} = 5$ )		Amazon Book ( $k^{\text{user}} = 30, k^{\text{item}} = 20$ )		LFM-1b ( $k^{\text{user}} = 300, k^{\text{item}} = 500$ )	
		NDCG (in %)	HR (in %)	NDCG (in %)	HR (in %)	NDCG (in %)	HR (in %)
Non Time-Aware	SASRec	58.27 $\pm$ 0.14	80.86 $\pm$ 0.46	55.23 $\pm$ 0.16	78.29 $\pm$ 0.19	<u>58.01 <math>\pm</math> 0.22</u>	73.84 $\pm$ 0.19
	BERT4Rec	53.75 $\pm$ 0.17	75.59 $\pm$ 0.18	57.17 $\pm$ 0.15	79.90 $\pm$ 0.08	56.08 $\pm$ 0.04	68.87 $\pm$ 0.06
	SSE-PT	55.62 $\pm$ 0.24	79.61 $\pm$ 0.19	55.00 $\pm$ 0.25	79.63 $\pm$ 0.11	56.60 $\pm$ 0.39	<u>75.21 <math>\pm</math> 0.36</u>
	AttRec	42.08 $\pm$ 0.31	69.23 $\pm$ 0.21	44.67 $\pm$ 0.20	71.00 $\pm$ 0.24	43.81 $\pm$ 0.19	64.66 $\pm$ 0.38
	FISSA	48.53 $\pm$ 0.31	74.16 $\pm$ 0.36	58.94 $\pm$ 0.11	81.42 $\pm$ 0.10	52.40 $\pm$ 0.19	68.20 $\pm$ 0.23
Time-Aware	TiSASRec	58.09 $\pm$ 0.26	<u>80.86 <math>\pm</math> 0.28</u>	52.16 $\pm$ 0.07	78.53 $\pm$ 0.22	55.67 $\pm$ 0.33	73.25 $\pm$ 0.37
	MEANTIME	<b>59.97 <math>\pm</math> 0.18</b>	79.76 $\pm$ 0.15	<u>58.95 <math>\pm</math> 0.11</u>	82.02 $\pm$ 0.11	57.03 $\pm$ 0.06	71.60 $\pm$ 0.07
	CARCA	38.65 $\pm$ 0.14	64.51 $\pm$ 0.09	56.82 $\pm$ 0.23	<u>82.89 <math>\pm</math> 0.17</u>	49.38 $\pm$ 0.44	66.30 $\pm$ 0.38
	MOJITO (ours)	<u>59.82 <math>\pm</math> 0.17</u>	<b>82.09 <math>\pm</math> 0.22</b>	<b>59.94 <math>\pm</math> 0.23</b>	<b>83.26 <math>\pm</math> 0.09</b>	<b>60.14 <math>\pm</math> 0.19</b>	<b>75.72 <math>\pm</math> 0.23</b>

## 4 EXPERIMENTAL ANALYSIS

This Section 4 presents our experimental evaluation of MOJITO. For reproducibility and future usage, we release our code on GitHub<sup>3</sup>.

### 4.1 Experimental Setting

**4.1.1 Datasets.** We consider three real-world datasets covering different domains, namely movie, book, and music recommendation:

- (1) MovieLens [12]: a movie recommendation dataset gathering 1 million ratings of 3 883 movies by 6 040 users. Each rating acts as a binary “interaction” in the following.
- (2) Amazon Book [22]: a collection of more than 1 million interactions with 96 421 books by 109 730 Amazon users.
- (3) LFM-1b [32]: more than a billion listening events by 120 322 Last.fm users, associated with 3 190 371 distinct music items.

Like Sun et al. [36], we implement a  $k^{\text{item}}$ -core (respectively, a  $k^{\text{user}}$ -core) pre-processing step on each dataset, consisting in recursively filtering data until all users (resp., all items) have at least  $k^{\text{item}}$  (resp.,  $k^{\text{user}}$ ) interactions. Values of  $k^{\text{item}}$  and  $k^{\text{user}}$  are reported in Table 1.

**4.1.2 Task.** For each interaction sequence, the last item is held out for test and the penultimate one for validation. We evaluate the ability of SR models to retrieve these items in evaluation sets containing the ground truth missing item and 1K other “negative” items that the user has not interacted with. SR models must recommend a list of 10 items, ordered by relevance scores. We evaluate recommendations using the Hit Rate (HR@10) and Normalized Discounted Cumulative Gain (NDCG@10) metrics. The former metric reports the percentage of top-10 lists containing missing items, whereas the latter also takes into account the missing item’s ranking in the list.

**4.1.3 Baselines.** We compare MOJITO to eight of the best Transformer models mentioned in Section 2.2: SASRec [17], BERT4Rec [35], SSE-PT [43], FISSA [15], AttRec [53], TiSASRec [20], MeanTime [7], and CARCA [30]. We recall that, like MOJITO, the three last baselines explicitly aim to capture temporal aspects for SR.

**4.1.4 Implementation Details.** We train all models for a maximum of 100 epochs using the Adam optimizer [18]. We set  $d = 64$ ,  $L = 50$ , batch sizes of 512, and  $H = 2$  for MOJITO. We selected all other hyperparameters via a grid search on validation items. For brevity,

we report optimal values of all models in our GitHub repository<sup>3</sup>. Most notably, we tested learning rates values in {0.0002, 0.0005, 0.00075, 0.001},  $N$  in {20, 50, 100}, and  $\lambda$  in {0.1, 0.2, 0.5, 0.8, 1.0}.

### 4.2 Results and Discussion

Table 1 reports all results on test sets, along with standard deviations computed over five model trainings. Overall, MOJITO reaches competitive results on all three datasets, which we discuss below.

**4.2.1 Time-Aware vs Non Time-Aware SR.** Our analysis reveals that time-aware systems tend to outperform non time-aware baselines, especially on MovieLens and Amazon Book. This highlights the relevance of modeling dependencies between preferences and the temporal context for SR in various domains. Yet, these performance differences are less pronounced on LFM-1b, likely due to the complexities of the temporal context in music consumption. Music consumption has lower levels of engagement than book or movie consumption, resulting in increased frequency of interactions and shorter time intervals between them [33]. This makes modeling the temporal context for music recommendation more challenging, especially given the higher number of music tracks for the same context. Nonetheless, MOJITO, explicitly accounting for day-of-the-week periodic information, has the best performance on LFM-1b (60.14% NDCG, 75.72% HR), showing its capacity to model the granularity of music consumption that is prone to periodic patterns [4].

**4.2.2 MOJITO vs Other Time-Aware Systems.** While accounting for the temporal context is crucial, the approach used to achieve this goal is equally important. In our experiments, MOJITO outperforms time-aware baselines according to 5 out of 6 scores (e.g., with a top 83.26% HR and a top 59.94% NDCG on Amazon Book) and reaches the second-best result for the 6<sup>th</sup> score (i.e., 59.82% NDCG on MovieLens, close to the top 59.97% NDCG obtained with MEANTIME on this dataset). These positive results emphasize the effectiveness of our flexible mixture-based probabilistic approach to automatically learn complex relationships between the temporal context and user preferences from various application domains.

**4.2.3 Future Analyses.** To conclude our discussion, we formulate two additional conjectures, that we plan to investigate in future research. Firstly, we believe that the inferior performance of CARCA with respect to time-aware alternatives on MovieLens and LFM-1B

<sup>3</sup><https://github.com/deezer/sigir23-mojito>

in Table 1 could also be explained by the relative complexity of this system. Indeed, CARCA uses an entire Transformer encoder-decoder architecture while others only employ the decoder part [30]. Secondly, we recall that, by design, MOJITO aims to reduce head redundancies, which our experiments confirm. For instance, we report a  $0.85 \pm 0.43$  mean pairwise  $L_2$  distance between head distances at the output layer for MOJITO on Amazon Book, vs.  $0.57 \pm 0.13$  for TiSASRec and  $0.76 \pm 0.59$  for CARCA (large values indicate less head redundancy). This head diversity seems to contribute to performance gains, as in the work of Nguyen et al. [24], although more ablation studies will be required in future work for confirmation.

## 5 CONCLUSION

In this paper, we proposed an effective Transformer for time-aware SR using mixtures of attention-based temporal context and item embedding representations. Backed by conclusive experiments on real-world datasets from various domains, we demonstrated its ability to thoroughly capture the impact of the temporal context on user preferences, enabling a more careful sequential modeling of user actions for SR. We also identified areas of analysis that, in the future, would help us better understand and improve our system.

## REFERENCES

- [1] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-Aware Recommender Systems. *Recommender Systems Handbook* (2011), 217–253.
- [2] Srinadh Bhojanapalli, Ayan Chakrabarti, Himanshu Jain, Sanjiv Kumar, Michal Lukasik, and Andreas Veit. 2021. Eigen Analysis of Self-Attention and its Reconstruction from Partial Computation. *arXiv preprint arXiv:2106.08823* (2021).
- [3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022.
- [4] Théo Bontempelli, Benjamin Chapus, François Rigaud, Mathieu Morlon, Marin Lorient, and Guillaume Salha-Galvan. 2022. Flow Moods: Recommending Music by Moods on Deezer. *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), 452–455.
- [5] Chao Chen, Haoyu Geng, Nianzu Yang, Junchi Yan, Daiyue Xue, Jianping Yu, and Xiaokang Yang. 2021. Learning Self-Modulating Attention in Continuous Time Space with Applications to Sequential Recommendation. *Proceedings of the 38th International Conference on Machine Learning* (2021), 1606–1616.
- [6] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. 2018. RecSys Challenge 2018: Automatic Music Playlist Continuation. *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), 527–528.
- [7] Sung Min Cho, Eunhyeok Park, and Sungjoo Yoo. 2020. MEANTIME: Mixture of Attention Mechanisms with Multi-Temporal Embeddings for Sequential Recommendation. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), 515–520.
- [8] Hanwen Du, Hui Shi, Pengpeng Zhao, Deqing Wang, Victor Sheng, Yanchi Liu, Guanfeng Liu, and Lei Zhao. 2022. Contrastive Learning with Bidirectional Transformers for Sequential Recommendation. *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (2022), 396–405.
- [9] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.
- [10] Lei Guo, Jinyu Zhang, Tong Chen, Xinhua Wang, and Hongzhi Yin. 2022. Reinforcement Learning-Enhanced Shared-Account Cross-Domain Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).
- [11] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and Sequential User Embeddings for Large-Scale Music Recommendation. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), 53–62.
- [12] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 1–19.
- [13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-Based Recommendations with Recurrent Neural Networks. *Proceedings of the 4th International Conference on Learning Representations* (2016).
- [14] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling Personalized Item Frequency Information for Next-Basket Recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), 1071–1080.
- [15] Lin Jing, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), 130–139.
- [16] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Factored Item Similarity Models for Top-N Recommender Systems. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2013), 659–667.
- [17] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. *Proceedings of the 20th International Conference on Data Mining* (2018), 197–206.
- [18] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations* (2015).
- [19] Cheng-Te Li, Cheng Hsu, and Yang Zhang. 2022. FairSR: Fairness-Aware Sequential Recommendation through Multi-Task Learning with Preference Graph Embeddings. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 1 (2022), 1–21.
- [20] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. *Proceedings of the 13th International Conference on Web Search and Data Mining* (2020), 322–330.
- [21] Xiaoyang Liu, Chong Liu, Pinzheng Wang, Rongqin Zheng, Lixin Zhang, Leyu Lin, Zhijun Chen, and Liangliang Fu. 2022. UFNRec: Utilizing False Negative Samples for Sequential Recommendation. *arXiv preprint arXiv:2208.04116* (2022).
- [22] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. *Proceedings of the 38th ACM SIGIR Conference on Research and Development in Information Retrieval* (2015), 43–52.
- [23] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better Than One? *Advances in Neural Information Processing Systems* 32 (2019).
- [24] Tan M. Nguyen, Tam Nguyen, Hai Do, Khai Nguyen, Vishwanath Saragadam, Minh Pham, Duy Khuong Nguyen, Nhat Ho, and Stanley J. Osher. 2022. Improving Transformer with an Admixture of Attention Heads. *Advances in Neural Information Processing Systems* 35 (2022).
- [25] Chanyoung Park, Donghyun Kim, Xing Xie, and Hwanjo Yu. 2018. Collaborative Translational Metric Learning. *Proceedings of the 2018 IEEE International Conference on Data Mining* (2018), 367–376.
- [26] Bruno L Pereira, Alberto Ueda, Gustavo Penha, Rodrygo LT Santos, and Nivio Ziviani. 2019. Online Learning to Rank for Sequential Music Recommendation. *Proceedings of the 13th ACM Conference on Recommender Systems* (2019), 237–245.
- [27] Aleksandr Petrov and Craig MacDonald. 2022. Effective and Efficient Training for Sequential Recommendation using Recency Sampling. *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), 81–91.
- [28] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), 579–588.
- [29] Massimo Quadrona, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Computing Surveys* 51, 4 (2018), 1–36.
- [30] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. Context and Attribute-Aware Sequential Recommendation via Cross-Attention. *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), 71–80.
- [31] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence* (2019), 4806–4813.
- [32] Markus Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval* (2016), 103–110.
- [33] Markus Schedl. 2019. Deep learning in music recommendation systems. *Frontiers in Applied Mathematics and Statistics* (2019), 44.
- [34] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current Challenges and Visions in Music Recommender Systems Research. *International Journal of Multimedia Information Retrieval* 7, 2 (2018), 95–116.
- [35] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), 1441–1450.
- [36] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), 23–32.
- [37] Yi Tay, Anh Tuan Luu, and Cheung Hui Siu. 2018. Latent Relational Metric Learning via Memory-Based Attention for Collaborative Ranking. *Proceedings of the 27th World Wide Web Conference* (2018), 729–739.
- [38] Viet-Anh Tran, Guillaume Salha-Galvan, Romain Hennequin, and Manuel Mousallam. 2021. Hierarchical Latent Relation Modeling for Collaborative Metric Learning. *Proceedings of the 15th ACM Conference on Recommender Systems* (2021),

- 302–309.
- [39] Xuan Tuan Trinh and Minh Phuong Tu. 2017. 3D Convolutional Networks for Session-Based Recommendation with Content Features. *Proceedings of the 11th ACM Conference on Recommender Systems* (2017), 138–146.
  - [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *Advances in Neural Information Processing Systems* 30 (2017), 5998–6008.
  - [41] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads do the Heavy Lifting, the Rest can be Pruned. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), 5797–5808.
  - [42] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Defu Lian. 2021. A Survey on Session-Based Recommender Systems. *ACM Computing Surveys* 54, 7 (2021), 1–38.
  - [43] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential Recommendation via Personalized Transformer. *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), 328–337.
  - [44] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence* (2019), 346–353.
  - [45] Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. 2022. Multi-Behavior Sequential Recommendation with Temporal Graph Transformer. *IEEE Transactions on Knowledge and Data Engineering* (2022).
  - [46] Xuefeng Xian, Ligang Fang, and Shiming Sun. 2021. Learning Heterogeneous Temporal Patterns of User Preference for Timely Recommendation. *Proceedings of the 30th Web Conference* (2021), 1274–1283.
  - [47] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering* (2022), 1259–1273.
  - [48] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-attention Network for Session-Based Recommendation. *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (2019), 3940–3946.
  - [49] Da Xu, Chuanwei Ruan, Sushant Kumar, Evren Korpeoglu, and Kannan Achan. 2019. Self-Attention with Functional Time Representation Learning. *Advances in Neural Information Processing Systems* 32 (2019).
  - [50] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-Behavior Hypergraph-Enhanced Transformer for Sequential Recommendation. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), 2263–2274.
  - [51] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time Matters: Sequential Recommendation with Complex Temporal Information. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), 1459–1468.
  - [52] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenberg, and Jure Leskovec. 2019. Hierarchical Temporal Convolutional Networks for Dynamic Recommender Systems. *Proceedings of the 28th World Wide Web Conference* (2019), 2236–2246.
  - [53] Shuai Zhang, Yi Tay, Lina Yao, and Aixin Sun. 2019. Next Item Recommendation with Self-Attentive Metric Learning. *AAAI 2019 Workshop on Recommender Systems and Natural Language Processing* (2019).
  - [54] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Junhua Fang, Lei Zhao, Victor S. Sheng, and Zhiming Cui. 2019. AdaCML: Adaptive Collaborative Metric Learning for Recommendation. *Proceedings of the 24th International Conference on Database Systems for Advanced Applications* (2019), 301–316.
  - [55] Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. 2022. Enhancing Sequential Recommendation with Graph Contrastive Learning. *Proceedings of the 31st International Joint Conference on Artificial Intelligence* (2022), 2398–2405.
  - [56] Chang Zhou, Jinze Bai, Junshuai Song, Xiaofei Liu, Zhengchao Zhao, Xiuxi Chen, and Jun Gao. 2018. ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (2018), 4564–4571.