INTERPOS: Interaction Rhythm Guided Positional Morphing for Mobile App Recommender Systems

M.H. Maqbool♣, Moghis Fereidouni♡, Umar Farooq♠, A.B. Siddique♡, Hassan Foroosh♢
hasan.ucf@gmail.com♣, moghis.fereidouni@uky.edu♡, ufarooq@lsu.edu♠, siddique@cs.uky.edu♡, hassan.foroosh@ucf.edu♢
Independent Researcher♣, *University of Kentucky*♡, *Louisiana State University*♠, *University of Central Florida*♢

***Abstract*—The mobile app market has expanded exponentially, offering millions of apps with diverse functionalities, yet research in mobile app recommendation remains limited. Traditional sequential recommender systems utilize the order of items in users' historical interactions to predict the next item for the users. Position embeddings, well-established in transformer-based architectures for natural language processing tasks, effectively distinguish token positions in sequences. In sequential recommendation systems, position embeddings can capture the order of items in a user's historical interaction sequence. Nevertheless, this ordering does not consider the time elapsed between two interactions of the same user (e.g., 1 day, 1 week, 1 month), referred to as "user rhythm". In mobile app recommendation datasets, the time between consecutive user interactions is notably longer compared to other domains like movies, posing significant challenges for sequential recommender systems. To address this phenomenon in the mobile app domain, we introduce INTERPOS, an Interaction Rhythm Guided Positional Morphing strategy for autoregressive mobile app recommender systems. INTERPOS incorporates rhythm-guided position embeddings, providing a more comprehensive representation that considers both the sequential order of interactions and the temporal gaps between them. This approach enables a deep understanding of users' rhythms at a fine-grained level, capturing the intricacies of their interaction patterns over time. We propose three strategies to incorporate the morphed positional embeddings in two transformer-based sequential recommendation system architectures. Our extensive evaluations show that INTERPOS outperforms state-of-the-art models using 7 mobile app recommendation datasets on NDCG@K and HIT@K metrics. The source code of INTERPOS is available at https://github.com/dlgrad/INTERPOS.**

***Index Terms*—recommendation systems, mobile app recommendations, sequential recommendations.**

I. Introduction

Mobile app market is witnessing exponential growth. Apple Appstore [1] and Google Play [2] include over 2.2 and 3.5 million apps, respectively [3]. Despite the vastness of the market and the abundance of apps accessible to users, the research in mobile app recommendation remains constrained.

Sequential recommender systems play a crucial role by leveraging the temporal order of items in users' historical interactions to predict their future preferences. In natural language processing tasks, position embedding is a well-established technique in transformer-based architectures to capture the order of tokens in sequences. However, in sequential recommendation systems, the reliance on position embeddings falls short in addressing a critical aspect – the temporal dynamics inherent in users' interaction patterns, which we refer to as "user rhythm". While position embedding can distinguish the order of items in a user's historical interactions, it does not take into account the time elapsed between two interactions by the same user, be it a day, a week, or a month. In mobile app recommendation datasets, the time gap between two consecutive user interactions is considerably longer compared to other widely studied domains, such as movies.

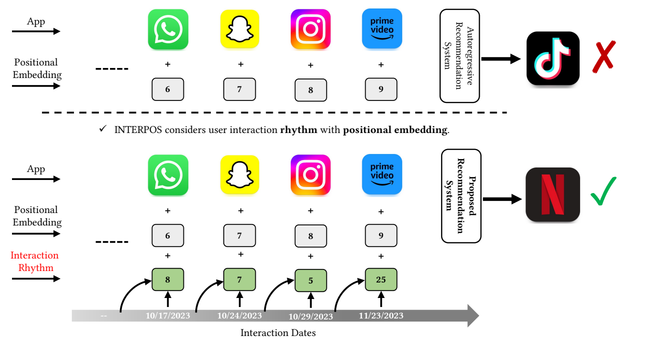


Fig. 1: INTERPOS considers user interaction rhythm with position embedding for tracking user preferences effectively reflecting the user's behavior over time.

To quantify this phenomenon, we analyze the time intervals between consecutive user interactions in mobile app recommendation datasets and compare them with other domains, including various categories from the AMAZON PRODUCT REVIEWS dataset [4]–[6] (e.g., Beauty, Video Games, CDs and Vinyl, Software, Grocery and Gourmet Food) and MOVIELENS-1M [7]. Our analysis, detailed in Figure 2, reveals an important pattern in other domain datasets: a substantial concentration of zero time differences between user interactions. Since many datasets, except for mobile app recommendation datasets, exhibit interactions on the same day, existing methods [8], [9] that consider time intervals between interactions do not need to learn to account for extended time gaps. These methods employ interval-based self-attention mechanisms, which are not effective in capturing the longer gaps between interactions.

In this work, we introduce a novel approach that integrates the gaps between successive interactions directly at the embedding layer. By embedding the time intervals early in the model architecture, we allow the model to learn and incorporate the temporal dynamics of user behavior more effectively. This approach captures longer gaps between interactions effectively,

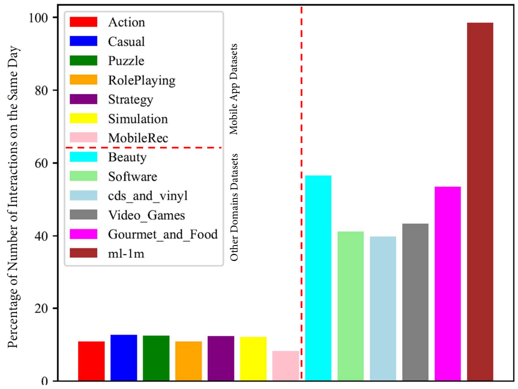


Fig. 2: Percentage of consecutive user interactions on the same day in the mobile app domain compared to other domains.

which are particularly prevalent in the mobile app domain, and significantly enhances the model's ability to generate accurate recommendations compared to existing techniques [8], [9] that fail to adequately account for these extended time intervals. Figure 1 provides an overview of our approach.

Specifically, we build INTERPOS on the observation that a user's interaction rhythm encodes valuable behavioral patterns that can be used for morphing the position encoding of items effectively. To achieve this, we introduce three fusion-based strategies for transformer-based recommender system architectures. These strategies, namely basic fusion, multilayer perceptron-based fusion, and gated fusion, are developed to integrate user rhythm into the recommendation process. The basic fusion strategy adds traditional position embeddings with rhythmic embeddings. The multilayer perceptron-based fusion incorporates additional layers of multilayer perceptrons after concatenating absolute position and user rhythm embeddings, enhancing the model's capacity to capture intricate relationships. Lastly, the gated fusion leverages a gating mechanism, enabling the model to learn how to judiciously mix and match positional embedding with rhythmic embedding.

To validate the efficacy of our proposed strategies, we implement them in two well-established transformer-based architectures: LightSANs [10] and SASRec [11]. Through this integration, we report considerable performance improvements across 7 mobile app recommendation datasets. On average, across all the datasets, INTERPOS fusion strategies, called INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF, achieve up to 157.3%, 156.67%, and 158% improvement in *NDCG@20* and 145.62%, 141.01%, and 142.85% improvement in *NDCG@20*, respectively. We observe that, across all the dataset splits, INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF show up to 145.15%, 139.39%, and 1143.63% improvement in *HIT@10* and 142.6%, 133.04%, and 137.04% improvement in *HIT@20*.

Our contributions are as follows:

- We investigate an underexplored domain in recommendation systems: mobile app recommendations.
- We introduce a novel paradigm that seamlessly integrates position embedding with user interaction rhythm by developing three fusion strategies for transformer-based recommender system architectures.
- We empirically demonstrate the effectiveness of incorporating interaction rhythm guidance and morphed positioning through extensive studies in mobile app recommendation, showcasing its superior performance.

II. Preliminaries

Problem Formulation. Let $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \ldots, \mathcal{U}_{|\mathcal{U}|}\}$ and $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \ldots, \mathcal{I}_{|\mathcal{I}|}\}$ represent the set of users and apps, respectively. A user-app interaction sequence is defined by $\mathcal{S}^{\mathcal{U}_i} = \left\{(\mathcal{U}_i, \mathcal{I}_p)_{|\mathcal{S}^{\mathcal{U}_i}|}\right\}$ where $i \in \{x \mid 1 \leq x \leq \mathcal{U}_{|U|}\}$ and $p \in \{y \mid 1 \leq y \leq \mathcal{I}_{|I|}\}$. Given a user-app interaction sequence for i-th user:

$$\mathcal{S}^{\mathcal{U}_i} = \left\{(\mathcal{U}_i, \mathcal{I}_1)_1, (\mathcal{U}_i, \mathcal{I}_2)_2, \ldots, (\mathcal{U}_i, \mathcal{I}_p)_{|\mathcal{S}^{\mathcal{U}_i}|-1}\right\} \tag{1}$$

up to sequence length $|\mathcal{S}^{\mathcal{U}_i}| - 1$, sequential recommendation seeks to predict the next app which a user is likely to interact with, $\mathcal{I}_{p+1}$ in the user's next interaction $(\mathcal{U}_i, \mathcal{I}_{p+1})_{|\mathcal{S}^{\mathcal{U}_i}|}$. It is important to note that the users are not explicitly modeled in the task formulation. The users are represented by their interaction history. Having noted this detail, we can simplify the notation by dropping the reference to the user in the interaction history, $\mathcal{S}^{\mathcal{U}_i}$. The simplified user-app interaction sequence can be written as:

$$\mathcal{S}^{\mathcal{U}_i} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \ldots, \mathcal{S}_{|\mathcal{S}^{\mathcal{U}_i}|}) \tag{2}$$

Given an input user/item sequence $\mathcal{S}^{\mathcal{U}_i} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \ldots, \mathcal{S}_{|\mathcal{S}^{\mathcal{U}_i}|-1})$ the model is expected to predict the shifted version of the input $\mathcal{S}^{\mathcal{U}_i} = (\mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5, \ldots, \mathcal{S}_{|\mathcal{S}^{\mathcal{U}_i}|})$. Sequential data can be modeled with an autoregressive transformers architecture [12]–[20].

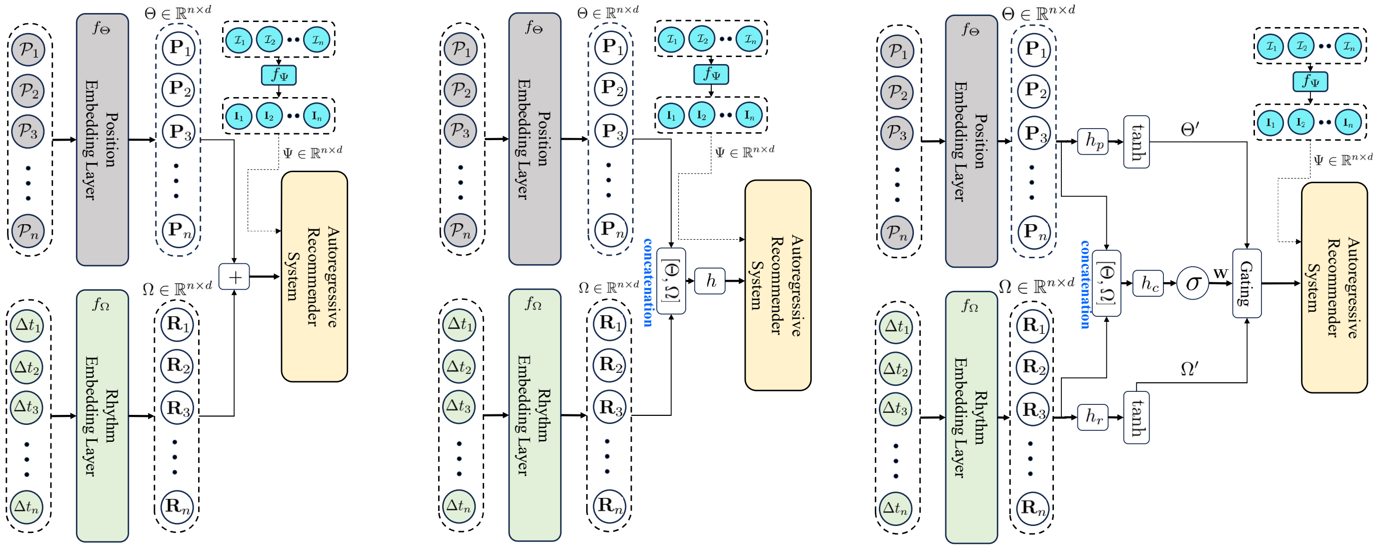
Self-attention. Self-attention is the main architectural choice behind transformers. Self-attention is defined as: $\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$, where Q, K, and V are query, key, and value matrices, respectively. In a transformer-based next-item prediction task, an embedding layer, f_Ψ, is employed to encode the items in a user's interaction history, $\mathcal{S}^{\mathcal{U}_i}$, into the embedding space.

$$\Psi = f_\Psi(\mathcal{S}^{\mathcal{U}_i}), \tag{3}$$

where item embedding matrix $\Psi \in \mathbb{R}^{n \times d}$ and d is the embedding dimension. Correspondingly, the input embedding matrix can be thought of as containing item embeddings.

Each $\mathbf{I}_i$ represents an item in the embedding space. The self-attention mechanism must be equipped with a sense of order in the input sequence. Let us define the absolute positions as $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_1, \ldots, \mathbf{P}_n\}$. $f_\Theta(.)$ is employed for encoding the absolute position information into embedding space, represented as Θ. Position embeddings have the same dimension as the input embeddings $\Theta \in \mathbb{R}^{n \times d}$. Finally, $E = \Psi + \Theta$ represents the user-item interaction sequence, $\mathcal{S}^{\mathcal{U}_i}$ in the embedding space:

$$\mathbf{E} = \begin{bmatrix} \mathbf{I}_1 + \mathbf{P}_1 \\ \mathbf{I}_2 + \mathbf{P}_2 \\ \vdots \\ \mathbf{I}_n + \mathbf{P}_n \end{bmatrix} \tag{4}$$



(a) INTERPOS-BF (b) INTERPOS-MF (c) INTERPOS-GF

Fig. 3: Overview of INTERPOS Fusion Architectures.

III. Approach

First, we conceptualize and represent the mathematical formulation of our proposed approach.

A. User's Interaction Rhythm

A user-item interaction is represented by a sequence $\mathcal{S}^{\mathcal{U}_i}$ for a user $\mathcal{U}_i$. We assume that a user's interaction sequence emerges across a particular period, and we call it the *Activity Window*. Referring back to Equation 1, we can rewrite the Equation 2 as follows:

$$\mathcal{S}^{\mathcal{U}_i} = \left(\mathcal{S}_1^{t_1}, \mathcal{S}_2^{t_2}, \mathcal{S}_3^{t_3}, \mathcal{S}_4^{t_4}, \ldots, \mathcal{S}_{|\mathcal{S}^{\mathcal{U}_i}|}^{t_N}\right) \tag{5}$$

where t_N represents the time of interaction for the last interaction for a user $\mathcal{U}_i$. Let us define $\Delta t_1 = 0$ and $\Delta t_i = t_i - t_{i-1} \forall i \in \{2, 3, \ldots, n\}$. Having the notion of Δt_i noted, let us define $f_\Omega(.)$ to encode a user's interaction rhythm Δt_i to the embedding space. Now, we can define the user's embedded interaction rhythm, Ω, as follows:

$$\Omega = \{\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \ldots, \mathbf{R}_n\} \tag{6}$$

Ω encodes the useful behavioral and preferential shifts for a user over the *Activity Window*. $\mathbf{R}_i$ represents the encoded Δt_i in the embedding space, i.e., $f_\Omega(\Delta t_i)$. We argue that Ω can be consumed by next-item prediction architectures for making more informed predictions. We propose three different architectural variations to fuse Ω into an autoregressive next-item prediction architecture.

B. Fusion Architectures

We present three different fusion methodologies to incorporate Ω into autoregressive next-item prediction architectures. Our proposed methodologies morph the existing position encoding Θ, by fusing Ω into it, effectively enriching the model's ability to discern the relative importance of the user's interaction history over the *Activity Window*.

Basic Fusion. Basic fusion employs a direct fusion approach of Θ and Ω. Let us define a vector-valued function $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$. f can represent element-wise summation, element-wise multiplication, or some other function. In our case, F is a simple element-wise addition. Let us say $\mathbf{M}$ represents a morphed version of Θ by fusing it with Ω through f:

$$\mathbf{M} = f(\Theta, \Omega) \tag{7}$$

$\mathbf{M}$ is employed to update Equation 4 to get the user-app interaction sequence. $\mathcal{S}^{\mathcal{U}_i}$ in the updated embedding space, $\mathbf{E}'$, with morphed position embeddings,

$$\mathbf{E}' = \begin{bmatrix} \mathbf{I}_1 + \mathbf{M}_1 \\ \mathbf{I}_2 + \mathbf{M}_2 \\ \vdots \\ \mathbf{I}_n + \mathbf{M}_n \end{bmatrix} \tag{8}$$

where $\mathbf{M}_i$ is an element-wise sum of $\mathbf{P}_i$ and $\mathbf{R}_i$. Figure 3a depicts the INTERPOS-BF architecture.

MLP Fusion. Let us define h to be a multi-layer perceptron such that $h : \mathbb{R}^{2d} \to \mathbb{R}^d$. Given Θ and Ω, $[\Theta, \Omega]$ represent their instance-wise concatenation, resulting in $\mathbb{R}^{n\times 2d}$ matrix. Having Θ and Ω concatenated, h is employed to generate the fused matrix $\mathbf{M}$,

$$\mathbf{M} = h([\Theta, \Omega]) \tag{9}$$

Equation 8 is employed for generating the updated embeddings for $\mathcal{S}^{\mathcal{U}_i}$ where $\mathbf{M}_i$ represents $\mathbf{h}([\mathbf{P}_i, \mathbf{R}_i])$ Figure 3b outlines the INTERPOS-MF fusion architecture.

Gated Fusion. Let h_p, h_r and h_c be multi-layer perceptrons. Θ is linearly projected using h_p and $\tanh$ non-linearity is applied such that $h_p(\Theta) \in \mathbb{R}^{n\times d}$. h_r is employed to project the rhythm embeddings Ω to the embedding space, followed by a $\tanh$ non-linearity such that $h_r(\Omega) \in \mathbb{R}^{n\times d}$. Gating matrix $\mathbf{W}$ is obtained by linearly projecting $[\Theta, \Omega] \in \mathbb{R}^{n\times 2d}$ using h_c where $[\Theta, \Omega] \in \mathbb{R}^{n\times 2d}$ represents the concatenation of $\Theta \in \mathbf{R}^{n\times d}$

TABLE I: Datasets statistics.

Descriptor ↓ Dataset →	Action	RolePlaying	Casual	Simulation	Strategy	Puzzle	MobileRec
# Unique Users	80961	77858	48091	56771	52055	51456	0.7 M
# Unique Apps	529	658	432	537	520	537	10173
Avg. interactions per user	9.82	9.63	8.22	8.58	8.37	8.34	27.56
Avg. interactions per app	1502.93	1139.57	915.47	906.56	838.06	798.92	1896.88
Maximum interactions by a user	29	30	23	24	22	24	256
Maximum interactions on an app	5372	3663	5209	3180	305	3742	14,345
Total Interactions (in Millions)	0.79 M	0.74 M	0.39 M	0.48 M	0.42 M	0.43 M	19.3 M

and $\Omega \in \mathbf{R}^{n\times d}$. Once $[\Theta, \Omega] \in \mathbb{R}^{n\times 2d}$ has been projected, a sigmoid function is employed to get the gating signal. Gated fusion can be represented as follows:

$$\Theta' = \tanh(h_p(\Theta)) \tag{10}$$

$$\Omega' = \tanh(h_r(\Omega)) \tag{11}$$

$$\mathbf{W} = \sigma(h_c([\Theta, \Omega])) \tag{12}$$

$$\mathbf{M} = \mathbf{W} \odot \Theta' + (1 - \mathbf{W}) \odot \Omega' \tag{13}$$

where Θ' and Ω' represent the linear projections of Θ and Ω. σ represents the *sigmoid* function, and $\mathbf{W}$ is the gating signal that is employed to fuse the updated position and user's interaction rhythm to get the fused matrix, $\mathbf{M} \in \mathbb{R}^{n\times d}$. Figure 3c illustrates the INTERPOS-GF architecture.

IV. Experiments

A. Experimental Setup

We use RecBole [26]–[28] for the implementation. We integrate INTERPOS into LightSANs [10] and SASRec [11] architectures. LightSANs integration is as follows. We use two transformer layers with two attention heads, the hidden size is 64 and the inner size is 256, while the number of latent interests is set to 5. The sequence length is capped at 50, with a hidden dropout probability of 0.5 and an attention dropout probability of 0.5. Hidden activation is Gaussian Error Linear Unit (GELU) and cross-entropy (CE) loss for the next-item prediction task. Models are trained for 100 epochs with an early-stopping patience of 10 with a batch size of 4096, and our learning rate is 0.001. SASRec has two transformer layers with two attention heads; the hidden size is 128 and the inner size is 256. We employ GELU activation and the *Adam* optimizer with a learning rate of 0.001. The same base SASRec architecture is consistently kept across for INTERPOS integration to get INTERPOS-BM, INTERPOS-GF, and INTERPOS-MF. The models' trainable parameters depend on the maximum interaction rhythm difference. A leave-one-out strategy is employed for validation and testing. The full item set is used for evaluation. We normalize the interaction rhythms in Action, Casual, RolePlaying, Puzzle, Simulation, Strategy with 0.2 and in case of MobileRec dataset, we clip the interaction rhythm at 800. We employ *HIT@K* and *NDCG@K* evaluation metrics where $k \in \{10, 15, 20\}$.

B. Datasets

Our experiments use 7 datasets for evaluating INTERPOS and benchmarking the performance gains in comparison with the competing baselines. MobileRec [29] is a large-scale dataset with over 19 million user-app interactions spanning 48 categories. We select the top 6 categories with the highest number of interactions, including (Action, RolePlaying, Puzzle, Casual, Simulation, Strategy, and Simulation). On top of these 6 datasets, a full-scale MobileRec [29] dataset with all 48 categories has also been used to establish the efficacy of INTERPOS. Table I presents detailed statistics of datasets.

C. Baselines

We have employed several strong baselines for benchmarking the efficacy of INTERPOS, as described in the following.

Pop is a simple popularity-based recommender system. This model captures the popularity of items in the dataset and suggests the most popular items to users as recommendations. **GRU4Rec** [21] applies an RNN-based method for the session-based recommendation. This baseline presents a method to account for data distribution shifts along with data augmentation. **LightSANs** [10] introduces a self-attention network with low-rank decomposition that projects users' historical items onto a small number of latent interests. **SASRec** [11] employs the attention mechanism for sequential recommendation task. **SINE** [22] proposes to use multiple embeddings to capture various aspects of a user's behavior. **HGN** [23] emphasizes the importance of recent chronological user-app interactions and integrates Bayesian Personalized Ranking (BPR) to capture both long-term and short-term user interests. **GCSAN** [24] a graph-contextualized self-attention model is proposed that employs both graph neural networks and self-attention networks. Graph neural network captures rich local associations, while self-attention networks capture long-range correlations. **BERT4Rec** [25] utilizes bidirectional self-attention, framing the sequential recommendation problem under the cloze objective. **TiSASRec** [8] incorporates time intervals between user interactions by explicitly modeling the timestamp of the interactions in the self-attention layer. **FEARec** [9] explicitly learns low-frequency and high-frequency information and combines time and frequency characteristics via auto-correlation.

V. Results and Discussion

We observe that INTERPOS with all the variations, exhibits a strong capability to steer the target model towards better performance. Our experiments also show that the proposed rhythm integration strategies INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF show their strengths on all datasets.

Performance on Action dataset. Action dataset has ≈81k unique users and 529 unique apps with nearly 1500 interactions per unique app and around 9 interactions per unique user on average. We report our results for Action dataset in Table II. Overall, when INTERPOS is incorporated in the LightSANs,

TABLE II: Results on Action dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0111	0.0132	0.0144	0.0248	0.0325	0.0377
Sequential	GRU4Rec [21]		0.0130	0.0166	0.0197	0.0295	0.0432	0.0565
	LightSANs [10]		0.0142	0.0181	0.0217	0.0312	0.0458	0.0609
	SASRec [11]		0.0134	0.0171	0.0202	0.0291	0.0432	0.0564
	SINE [22]		0.0091	0.0119	0.0142	0.0202	0.0307	0.0403
	HGN [23]		0.0128	0.0165	0.0199	0.0283	0.0425	0.0569
	GCSAN [24]		0.0129	0.0164	0.0195	0.0281	0.0412	0.0546
	BERT4Rec [25]		0.0096	0.0128	0.0154	0.0218	0.0339	0.0452
Time-aware	FEARec [9]		0.0147	0.0194	0.0201	0.0293	0.0461	0.0514
	TiSASRec [8]		0.0150	0.0199	0.0212	0.033	0.0514	0.0575
This work	**LightSANs**	**INTERPOS-BF**	0.0386	**0.0465**	**0.0533**	**0.0809**	**0.1107**	**0.1395**
		INTERPOS-GF	0.0385	0.0461	0.0523	0.0790	0.1077	0.1340
		INTERPOS-MF	**0.0387**	0.0463	0.0527	0.0804	0.1095	0.1363
	SASRec	**INTERPOS-BF**	0.0234	0.0286	0.0331	0.0491	0.069	0.0878
		INTERPOS-GF	0.0257	0.0316	0.0364	0.0539	0.0761	0.0964
		INTERPOS-MF	0.0262	0.0324	0.0375	0.0555	0.0788	0.1008

TABLE III: Results on RolePlaying dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0056	0.0084	0.0130	0.0132	0.0237	0.0436
Sequential	GRU4Rec [21]		0.0174	0.0210	0.0246	0.0362	0.0497	0.0651
	LightSANs [10]		0.0165	0.0206	0.0238	0.0357	0.0511	0.0649
	SASRec [11]		0.0146	0.018	0.0212	0.0522	0.0717	0.0916
	SINE [22]		0.0094	0.0112	0.0131	0.0192	0.0262	0.0343
	HGN [23]		0.0126	0.0160	0.0190	0.0270	0.0399	0.0523
	GCSAN [24]		0.0120	0.0149	0.0175	0.0258	0.0365	0.0477
	BERT4Rec [25]		0.0102	0.0128	0.0151	0.0210	0.0308	0.0403
Time-aware	FEARec [9]		0.0181	0.0191	0.024	0.04	0.0558	0.0651
	TiSASRec [8]		0.0194	0.0206	0.0203	0.0387	0.0522	0.0654
This work	**LightSANs**	**INTERPOS-BF**	0.0328	0.0398	0.0451	0.0684	0.0949	0.1174
		INTERPOS-GF	**0.0353**	**0.0429**	**0.0485**	**0.0747**	**0.1035**	**0.1271**
		INTERPOS-MF	0.0341	0.0412	0.0466	0.0724	0.0996	0.1220
	SASRec	**INTERPOS-BF**	0.0252	0.0303	0.0350	0.0522	0.0717	0.0916
		INTERPOS-GF	0.0294	0.0354	0.0402	0.0612	0.084	0.1042
		INTERPOS-MF	0.0283	0.0342	0.0387	0.0597	0.0821	0.1012

it performs better than SASRec integration. In LightSANs integration, INTERPOS-BF performs better on *NDCG@k* where $k \in \{15, 20\}$. On *HIT@k*, INTERPOS-BF is a better-performing strategy for $k \in \{10, 15, 20\}$. This observation that within one dataset, our proposed fusion strategies find their strengths on different values of k in both *NDCG* and *HIT* can be attributed to relative metric strictness. We argue that the choice of INTERPOS (i.e., INTERPOS-BF, INTERPOS-GF, INTERPOS-MF) fusion strategies depends upon the dataset's distributional variations and the underlying metric. In comparison with other competing baselines, we observe that INTERPOS outperforms all the baselines by a significant margin. Among the baselines, time-aware baselines perform the best while the popularity remains the worst. The sequential recommendation baselines have mixed results among all the sequential baselines. *LightSANs* [10] demonstrates strong performance on *NDCG@k* and *HIT@k* for all values of *k*. The time-aware baselines outperform all the sequential baselines except for the *NDCG@20* metric, which shows that time-sensitivity in fact plays an important role. INTERPOS manages to outperform time-aware baselines by a large margin. As compared to best-performing baselines, on *NDCG@10*, we observe a percentage improvement of 157.3%, 156.67%, and 158% by INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF respectively in *LightSANs* integration. On *NDCG@20*, INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF manage to achieve 145.62%, 141.01%, and 142.85% improvement compared to *LightSANs* [10], respectively. On *HIT@10* and *HIT@20*, INTERPOS-BF manages a percentage improvement of 145.15% and 142.6%, INTERPOS-GF demonstrates an improvement of 139.39% and 133.04% while INTERPOS-MF shows 143.63% and 137.04% improvement.

Performance on RolePlaying dataset. RolePlaying has roughly 78k unique users and 658 unique apps, with an average interaction of 1139.57 per unique app and 9.63 per unique user. Table III reports the results for RolePlaying. INTERPOS-MF is the best-performing fusion strategy among INTERPOS when integrated into *LightSANs*. Again, time-aware baselines perform better as compared to other baselines. Among sequential baselines, *GRU4Rec* [21] is the among all the baselines. INTERPOS-MF integration with *LightSANs* obtains a considerable percentage improvement over all baselines. INTERPOS-GF integration with *LightSANs* achieves a significant 88.62% improvement against *GRU4Rec* [21] on *NDCG@10*. Similarly, it demonstrates an 81.95% improvement on *NDCG@10* compared to *TiSASRec*. On *HIT@15* and *HIT@20*, INTERPOS-GF shows 85.48% and 94.34% improvement in comparison with time-aware baselines.

TABLE IV: Results on Casual dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0112	0.0144	0.0186	0.0204	0.0324	0.0501
Sequential	GRU4Rec [21]		0.0294	0.0351	0.0399	0.0576	0.0791	0.0998
	LightSANs [10]		0.0294	0.0355	0.0403	0.0559	0.0793	0.0996
	SASRec [11]		0.0283	0.0340	0.0386	0.0566	0.0783	0.0975
	SINE [22]		0.0256	0.0295	0.0327	0.0466	0.0613	0.0752
	HGN [23]		0.0288	0.0343	0.0392	0.0569	0.0778	0.0985
	GCSAN [24]		0.0267	0.0317	0.0361	0.0508	0.0699	0.0885
	BERT4Rec [25]		0.0252	0.0295	0.0331	0.0515	0.0678	0.0830
Time-aware	FEARec [9]		0.0304	0.041	0.0353	0.0599	0.089	0.0973
	TiSASRec [8]		0.0261	0.04	0.0344	0.057	0.0822	0.1023
This work	**LightSANs**	**INTERPOS-BF**	0.0390	0.0474	0.0551	0.0790	0.1108	0.1434
		INTERPOS-GF	0.0400	0.0484	0.0545	0.0817	0.1136	0.1394
		INTERPOS-MF	**0.0424**	**0.0518**	**0.0592**	**0.0871**	**0.123**	**0.1541**
	SASRec	**INTERPOS-BF**	0.0316	0.0377	0.0435	0.0634	0.0867	0.1113
		INTERPOS-GF	0.0343	0.0402	0.0455	0.0664	0.0891	0.1112
		INTERPOS-MF	0.0329	0.0392	0.0445	0.0654	0.0894	0.1119

TABLE V: Results on Strategy dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0194	0.0213	0.0244	0.0341	0.0411	0.0543
Sequential	GRU4Rec [21]		0.0212	0.0270	0.0313	0.0478	0.0698	0.0880
	LightSANs [10]		0.0268	0.0332	0.0384	0.0579	0.082	0.1040
	SASRec [11]		0.0186	0.0240	0.0285	0.0412	0.0618	0.0808
	SINE [22]		0.0098	0.0127	0.0156	0.0200	0.0308	0.0432
	HGN [23]		0.0207	0.0265	0.0312	0.0445	0.0664	0.0863
	GCSAN [24]		0.0171	0.0216	0.0255	0.0387	0.0559	0.0722
	BERT4Rec [25]		0.0140	0.0178	0.0208	0.0307	0.0454	0.0580
Time-aware	FEARec [9]		0.0237	0.0336	0.0336	0.0599	0.0899	0.0943
	TiSASRec [8]		0.0253	0.0375	0.0375	0.0592	0.0912	0.1056
This work	**LightSANs**	**INTERPOS-BF**	**0.0403**	**0.0482**	**0.0552**	**0.086**	**0.116**	**0.1457**
		INTERPOS-GF	0.0395	0.0471	0.0540	0.0845	0.1133	0.1423
		INTERPOS-MF	0.0384	0.0465	0.0533	0.0816	0.1122	0.1411
	SASRec	**INTERPOS-BF**	0.0283	0.0348	0.0404	0.0611	0.0857	0.1093
		INTERPOS-GF	0.0299	0.0368	0.0426	0.064	0.0902	0.1150
		INTERPOS-MF	0.0316	0.0383	0.0442	0.0661	0.0914	0.1165

Performance on Casual dataset. Table IV summarizes the results on Casual dataset. INTERPOS-MF integrated with *LightSANs* outclasses baselines and other fusion strategies. Time-aware baselines perform better as compared to others with the exception of *NDCG@20* metric, where *LightSANs* [10] performs better as compared to other baselines. INTREPOS-MF integrated with *LightSANs* manages to outperform the best time-aware baseline (i.e., *FEARec*) by 39.47% on *NDCG@10* metric and by 67.70% on *NDCG@20*. Notable improvements are obtained by other fusion strategies on *NDCG@10* and *NDCG@15*. Consider the *NDCG@10* and *NDCG@15* metrics, INTERPOS-BF manages to outperform *FEARec* by 28.28% and 15.60% while INTERPOS-GF demonstrates a percent improvement of 31.57% and 18.04%. On *HIT@10*, *HIT@15*, and *HIT@20*, INTERPOS-MF shows 45.40%, 38.20%, and 50.63% improvements against the best-performing (i.e., time-aware) baselines on these metrics. INTERPOS-BF shows an improvement of 31.88%, 24.49%, and 40.17% on these metrics in comparison with the best-performing baselines. Similarly, INTERPOS-GF manages to obtain, 36.39%, 27.64%, and 36.26% improvement in comparison with the best performing baselines on *HIT@10*, and *HIT@15*, and *HIT@20*.

Performance on Simulation dataset. Table VI presents the results on Simulation dataset. On *NDCG@20*, under LightSANs integration, INTERPOS-BF demonstrates 72.94% improvement over the best-performing baseline *LightSANs*. INTERPOS-GF and INTERPOS-MF show 46.91% and 62.67% improvement over *LightSANs* on *NDCG@20*. On *NDCG@10 FEARec* outclasses other baselines. INTERPOS-BF manages to get 46.93% improvement over the best-performing baseline. On *HIT@10* and *HIT@20*, INTERPOS-BF outperforms the best baseline method *TiSASRec* by 52.52% and 62.67%, while INTERPOS-GF manages a 30.32% and 41.78% improvement and INTERPOS-MF shows a 48.13% and 57.74% improvement, respectively.

Performance on Strategy dataset. On Strategy dataset, INTERPOS in both integration with *LightSANs* and *SASRec* outclass all the competing baselines. Detailed results are shown in Table V. On *NDCG@10*, *LightSANs* [10] outclasses the other baseline methods. INTERPOS-BF under *LightSANs* integration manages to outperform *LightSANs* [10] on *NDCG@10* by 50.37%. Similarly, on *HIT@15* and *HIT@20*, INTERPOS-BF manages to outperform the best competing baselines *FEARec* and *LightSANs* [10] by 28.53% and 43.75%. On the other hand, on *NDCG@10*, INTERPOS-GF and INTERPOS-MF show a percentage improvement of 47.38%

TABLE VI: Results on Simulation dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0136	0.0178	0.0202	0.0255	0.0414	0.0518
Sequential	GRU4Rec [21]		0.0197	0.0245	0.0289	0.0407	0.0587	0.0774
	LightSANs [10]		0.0198	0.0248	0.0292	0.0412	0.0601	0.0789
	SASRec [11]		0.0175	0.0221	0.0263	0.0383	0.0558	0.0732
	SINE [22]		0.0145	0.0171	0.0200	0.0345	0.0445	0.0567
	HGN [23]		0.0181	0.0231	0.0273	0.0383	0.0571	0.0751
	GCSAN [24]		0.0179	0.0226	0.0268	0.0397	0.0579	0.0755
	BERT4Rec [25]		0.0149	0.0194	0.0227	0.0324	0.0495	0.0633
Time-aware	FEARec [9]		0.0245	0.0246	0.025	0.0417	0.0644	0.0731
	TiSASRec [8]		0.0159	0.0313	0.0285	0.0455	0.0611	0.0781
This work	**LightSANs**	**INTERPOS-BF**	**0.036**	**0.0435**	**0.0505**	**0.0694**	**0.0977**	**0.1277**
		INTERPOS-GF	0.0299	0.0369	0.0429	0.0593	0.0857	0.1113
		INTERPOS-MF	0.0335	0.0409	0.0475	0.0674	0.0953	0.1232
	SASRec	**INTERPOS-BF**	0.0177	0.0223	0.0261	0.0389	0.0564	0.0724
		INTERPOS-GF	0.0216	0.0274	0.0327	0.0462	0.0680	0.0905
		INTERPOS-MF	0.0211	0.0266	0.0316	0.0446	0.0657	0.0867

TABLE VII: Results on Puzzle dataset, best results are shown in bold, second best results are underlined.

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0172	0.0215	0.0244	0.0382	0.0545	0.0668
Sequential	GRU4Rec [21]		0.0224	0.0271	0.0317	0.0458	0.0635	0.0832
	LightSANs [10]		0.0229	0.0278	0.0322	0.0464	0.0652	0.0837
	SASRec [11]		0.0221	0.0270	0.0313	0.0446	0.0635	0.0816
	SINE [22]		0.0194	0.0227	0.0257	0.0377	0.0506	0.0629
	HGN [23]		0.0222	0.0270	0.0314	0.0454	0.0638	0.0823
	GCSAN [24]		0.0227	0.0274	0.0313	0.0465	0.0644	0.0812
	BERT4Rec [25]		0.0189	0.0228	0.0259	0.0363	0.0510	0.0642
Time-aware	FEARec [9]		0.0267	0.0288	0.0338	0.052	0.0679	0.0725
	TiSASRec [8]		0.0262	0.0316	0.0355	0.048	0.0717	0.0828
This work	**LightSANs**	**INTERPOS-BF**	0.0377	0.0444	0.0505	0.0719	0.0976	0.1234
		INTERPOS-GF	**0.0408**	**0.0485**	**0.0556**	**0.0789**	**0.1082**	**0.1379**
		INTERPOS-MF	0.0370	0.0440	0.0506	0.0701	0.0967	0.1248
	SASRec	**INTERPOS-BF**	0.0231	0.0285	0.0330	0.0478	0.0683	0.0872
		INTERPOS-GF	0.0283	0.0344	0.0401	0.0583	0.0814	0.1055
		INTERPOS-MF	0.0246	0.0303	0.0354	0.0515	0.0731	0.0949

and 43.28%, respectively. On *NDCG@20*, INTERPOS-GF and INTERPOS-MF manage to get 40.62% and 38.8% improvement over *LightSANs* [10]. On *HIT@10*, we notice that INTERPOS-BF demonstrates considerable improvement by 43.57% over *FEARec*, while INTERPOS-GF and INTERPOS-MF demonstrate 41.06% and 36.22% improvements over the best baseline. Similarly, a 27.19% and 37.97% improvement is shown by INTERPOS-BF on *HIT@15* and *HIT@20* evaluation metrics over *TiSASRec*.

Performance on Puzzle dataset. On Puzzle dataset, INTERPOS-GF turns out to be the outstanding fusion strategy. In comparison with the most competitive baseline *FEARec* on *NDCG@10*, INTERPOS-GF shows 52.80% improvement. Similarly, INTERPOS-GF obtains a performance improvement of 53.48% and 56.61% over the best-performing baseline (i.e., *TiSASRec*) on *NDCG@15* and *NDCG@20*, respectively. INTERPOS-BF gets 42.25% and INTERPOS-MF manages a 42.53% improvement over *TiSASRec* on *NDCG@20*. INTERPOS outclasses the best baseline on *HIT@10* and *HIT@20* by significant margins, where INTERPOS-GF shows the highest improvement among the INTERPOS variations, 51.73% and 64.75%, respectively.

Performance on MobileRec dataset. On the MobileRec dataset, we observe INTERPOS with SASRec settings are the best-performing models, INTERPOS-GF outperforms all of our variants on all metrics with the exception of *HIT@20*. We believe the key reason behind the INTERPOS integrated with SASRec variants outperforming LightSANs integrated INTERPOS variants is that MobileRec has the least percentage of user interactions on the same day as compared to other datasets (shown in Figure 2). Please see Table VIII for detailed results. In the MobileRec dataset, there are 19 million user-app interactions, 0.7 million unique users, and more than 10k unique applications. There are 27.56 interactions per user on average, while on average, there are more than 1800 interactions per unique app. On *NDCG@10*, *NDCG@15*, and *NDCG@20*, INTERPOS-GF demonstrates 8.04%, 7.76% and 9.48% improvement compared to the best competing baseline *SASRec*. On *HIT@10*, *HIT@15*, and *HIT@20*, INTERPOS varitants outclass the best-performing baseline *SASRec* by 12.20%, 10.30%, and 13.88%, respectively.

Summary of Results. In conclusion, we observe that often existing time-aware baselines, FEARec and TiSASRe, perform better than sequential baselines, which highlights the significance of incorporating time sensitivity into the models. However, these time-aware baselines incorporated time

TABLE VIII: Results on MobileRec dataset, best results are shown in bold, second best results are underlined

Category	Method ↓ Metric →		NDCG @10	NDCG @15	NDCG @20	HIT @10	HIT @15	HIT @20
Popularity	Pop		0.0077	0.0092	0.0103	0.0151	0.0208	0.0256
Sequential	GRU4Rec [21]		0.0074	0.0089	0.0102	0.0153	0.021	0.0261
	LightSANs [10]		0.0079	0.0092	0.0104	0.0158	0.0208	0.0260
	SASRec [11]		0.0087	0.0103	0.0116	0.0172	0.0233	0.0288
	SINE [22]		0.0076	0.0091	0.0102	0.0157	0.0213	0.0260
	HGN [23]		0.0046	0.0056	0.0064	0.0096	0.0132	0.0165
	GCSAN [24]		0.0081	0.0095	0.0107	0.0161	0.0214	0.0266
	BERT4Rec [25]		0.0069	0.0081	0.0091	0.0139	0.0185	0.0226
Time-aware	FEARec [9]		0.0077	0.0089	0.0071	0.0169	0.0221	0.0231
	TiSASRec [8]		0.0063	0.0091	0.0107	0.0167	0.0219	0.0259
This work	**LightSANs**	**INTERPOS-BF**	0.0083	0.0098	0.0111	0.0167	0.0224	0.0279
		INTERPOS-GF	0.0091	0.0107	0.0120	0.0186	0.0246	0.0301
		INTERPOS-MF	0.0082	0.0097	0.0110	0.0167	0.0224	0.0278
	SASRec	**INTERPOS-BF**	0.0089	0.0108	<u>0.0125</u>	0.0184	<u>0.0256</u>	**0.0328**
		INTERPOS-GF	**0.0094**	**0.0111**	**0.0127**	**0.0193**	**0.0257**	<u>0.0322</u>
		INTERPOS-MF	<u>0.0093</u>	**0.0111**	<u>0.0125</u>	<u>0.0191</u>	<u>0.0256</u>	0.0318

sensitivity at a later stage in contrast to INTERPOS, which fuses this information earlier on and produces significantly better results. Empirical results show that there is no clear winner in the fusion strategies. Yet, all fusion strategies exhibit statistically significant performance compared to all the baselines including sequential and time-aware recommendation systems across all 7 datasets. Moreover, fusion strategies integrated into *SASRec* demonstrate superior performance on MobileRec, while these strategies show better results on all other datasets when incorporated into *LightSANs*. It is also important to highlight despite not being a clear winner in the fusion strategies, the results of these strategies are statistically insignificant when compared with with another. We empirically show that INTERPOS can better capture the user's behavioral and preferential shifts over the *Activity Window*. Through these results, we demonstrate that the integration of the user's interaction rhythm into an autoregressive next-item prediction model can facilitate better learning of the user's behavioral pattern leading to tailored predictions.

VI. Related Work

Recommender systems have been applied in diversified domains like products [30], [31], news [32], movies [33] and a large body of research work focuses on improving recommender systems, which is the focus of this work. SASRec [11] strives to find a balance between Markov Chain (MC) based methods and Recurrent Neural Network (RNN) based designs. BERT4Rec [25] proposed to adopt a cloze objective for randomly masked items prediction by leveraging their bidirectional contextual conditioning. [34] proposes a knowledge-enhanced memory network using RNN with key-value memory networks.

He and others [35] fused similarity-based methods with Markov chains for personalized sequential recommendations. Observing that session embeddings and item embeddings are not in the same embedding space, [36] proposes a framework to unify the representation spaces for encoding and decoding processes. [37] proposes a model to make an informed decision on the consumption of repeated items. A short-term attention memory priority model is proposed in [38] [39] suggested extracting the self-supervision signal by utilizing the inherent correlation in the data to improve the data representation through pertaining. Convolutional filters are employed in [40] [41] proposes parallel RNNs to exploit user clicks and accompanying features (visual and textural) for modeling user interaction sessions. [42] proposes a two-layer hierarchical attention network to make use of the user's long-term historical interactions and short-term preferences. In comparison, our work fuses the user's interaction rhythm to improve sequential recommendation systems.

Yuan and others [43] presented the idea of a holed convolutional neural network. [44] argues to employ explicit features on top of the transition patterns of items Graph neural networks are employed in [45] A neural personalized embedding model is proposed in [46] for improving the recommendation performance for cold users. The authors in [47] proposed to employ transformer pretraining on reverse sequences and obtain predictions on the prior items. [8] considers the user's interaction timestamp within the sequence modeling. [48], [49], [50] study app reviews in context of developer responses and app issues.

VII. Conclusion and Future Work

In this work, we propose INTERPOS which employs a user's interaction rhythm to morph the position encodings to inject a sense of the user's behavior and preference shifts over the user's interactions in the context of mobile apps recommendation. We empirically establish that interaction rhythm is correlated with a user's behavioral pattern and can provide an autoregressive model with a unique perspective on a user's preference shifts. We incroporate three strategies, INTERPOS-BF, INTERPOS-GF, and INTERPOS-MF, into two transformer-based recommendation system architecture. We show that the proposed integration strategies show strong learning capacity and help improve the underlying autoregressive model for the mobile apps recommendation task. We empirically establish that INTERPOS variants outclass the sequential and time-aware baselines across all the datasets by a large margin. In the future, we will explore alternative strategies to incorporate the user's interaction rhythm into an autoregressive design.

References

[1] Apple, "Apple app store," https://apps.apple.com/, 2024, accessed: 2024-01-06.

[2] Google, "Google play store," https://play.google.com/store/apps, 2022, accessed: 2024-01-06.

[3] L. Ceci, "Number of apps available in leading app store," https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, accessed: 2024-01-06.

[4] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[5] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507–517.

[6] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 188–197.

[7] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[8] J. Li, Y. Wang, and J. McAuley, "Time interval aware self-attention for sequential recommendation," in *Proceedings of the 13th international conference on web search and data mining*, 2020, pp. 322–330.

[9] X. Du, H. Yuan, P. Zhao, J. Qu, F. Zhuang, G. Liu, Y. Liu, and V. S. Sheng, "Frequency enhanced hybrid attention network for sequential recommendation," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 78–88.

[10] X. Fan, Z. Liu, J. Lian, W. X. Zhao, X. Xie, and J.-R. Wen, "Lighter and better: low-rank decomposed self-attention networks for next-item recommendation," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 1733–1737.

[11] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[14] A. Roberts, C. Raffel, K. Lee, M. Matena, N. Shazeer, P. J. Liu, S. Narang, W. Li, and Y. Zhou, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019.

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

[16] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[17] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," *arXiv preprint arXiv:1909.05858*, 2019.

[18] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," *Advances in neural information processing systems*, vol. 32, 2019.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[20] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.

[21] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 17–22.

[22] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 598–606.

[23] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 825–833.

[24] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation." in *IJCAI*, vol. 19, 2019, pp. 3940–3946.

[25] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[26] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian, Y. Min, Z. Feng, X. Fan, X. Chen, P. Wang, W. Ji, Y. Li, X. Wang, and J. Wen, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *CIKM*. ACM, 2021, pp. 4653–4664.

[27] L. Xu, Z. Tian, G. Zhang, J. Zhang, L. Wang, B. Zheng, Y. Li, J. Tang, Z. Zhang, Y. Hou, X. Pan, W. X. Zhao, X. Chen, and J.-R. Wen, "Towards a more user-friendly and easy-to-use benchmark library for recommender systems," 2023, p. 2837–2847.

[28] W. X. Zhao, Y. Hou, X. Pan, C. Yang, Z. Zhang, Z. Lin, J. Zhang, S. Bian, J. Tang, W. Sun *et al.*, "Recbole 2.0: Towards a more up-to-date recommendation library," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4722–4726.

[29] M. H. Maqbool, U. Farooq, A. Mosharrof, A. B. Siddique, and H. Foroosh, "Mobilerec: A large scale dataset for mobile apps recommendation," p. 3007–3016, 2023. [Online]. Available: https://doi.org/10.1145/3539618.3591906

[30] A. Yan, C. Dong, Y. Gao, J. Fu, T. Zhao, Y. Sun, and J. McAuley, "Personalized complementary product recommendation," in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 146–151.

[31] M. M. Tanjim, C. Su, E. Benjamin, D. Hu, L. Hong, and J. McAuley, "Attentive sequential models of latent intent for next item recommendation," in *Proceedings of The Web Conference 2020*, 2020, pp. 2528–2534.

[32] F. Wu, Y. Qiao, J.-H. Chen, C. Wu, T. Qi, J. Lian, D. Liu, X. Xie, J. Gao, W. Wu *et al.*, "Mind: A large-scale dataset for news recommendation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3597–3606.

[33] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars)," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 193–202.

[34] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *The 41st international ACM SIGIR conference on research & development in information retrieval*, 2018, pp. 505–514.

[35] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 2016, pp. 191–200.

[36] Y. Hou, B. Hu, Z. Zhang, and W. X. Zhao, "Core: simple and effective session-based recommendation within consistent representation space," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1796–1801.

[37] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. De Rijke, "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4806–4813.

[38] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1831–1839.

[39] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1893–1902.

[40] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[41] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 241–248.

[42] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention network," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[43] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 582–590.

[44] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, X. Zhou *et al.*, "Feature-level deeper self-attention network for sequential recommendation." in *IJCAI*, 2019, pp. 4320–4326.

[45] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 346–353.

[46] T. Nguyen and A. Takasu, "Npe: neural personalized embedding for collaborative filtering," *arXiv preprint arXiv:1805.06563*, 2018.

[47] Z. Liu, Z. Fan, Y. Wang, and P. S. Yu, "Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer," in *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*, 2021, pp. 1608–1612.

[48] U. Farooq, A. B. Siddique, F. Jamour, Z. Zhao, and V. Hristidis, "App-aware response synthesis for user reviews," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 699–708.

[49] M. Fereidouni, A. Mosharrof, U. Farooq, and A. Siddique, "Proactive prioritization of app issues via contrastive learning," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 535–544.

[50] S. Maji, M. Fereidouni, V. Chhetri, U. Farooq, and A. B. Siddique, "Mobileconvrec: A conversational dataset for mobile apps recommendations," 2024. [Online]. Available: https://arxiv.org/abs/2405.17740