Hi-Fi Ark: Deep User Representation via High-Fidelity Archive Network

Zheng Liu[1] , **Yu Xing**[2] , **Fangzhao Wu**[1] , **Mingxiao An**[2] , **Xing Xie**[1]
[1]Microsoft Research Asia, Beijing, China
[2]University of Science and Technology China, Hefei, China
{zhengliu, v-yuxing, fangzwu, v-minan, xingx}@microsoft.com

Abstract

Deep learning techniques have been widely applied in modern recommendation systems, leading to flexible and effective ways of user representation. Conventionally, user representations are generated purely in the offline stage. Without referencing to the specific candidate item for recommendation, it is difficult to fully capture user preference from the perspective of interest. More recent algorithms tend to generate user representation at runtime, where user's historical behaviors are attentively summarized w.r.t. the presented candidate item. In spite of the improved efficacy, it is too expensive for many real-world scenarios because of the repetitive access to user's entire history. In this work, a novel user representation framework, Hi-Fi Ark, is proposed. With Hi-Fi Ark, user history is summarized into highly compact and complementary vectors in the offline stage, known as archives. Meanwhile, user preference towards a specific candidate item can be precisely captured via the attentive aggregation of such archives. As a result, both deployment feasibility and superior recommendation efficacy are achieved by Hi-Fi Ark. The effectiveness of Hi-Fi Ark is empirically validated on three real-world datasets, where remarkable and consistent improvements are made over a variety of well-recognized baseline methods.

1 Introduction

The flourish of deep learning techniques has greatly promoted the progress of recommendation systems. One of the most notable practices is user representation, where deep neural networks are employed to extract users' stable behavioral patterns in the long run. For such kind of applications, neural networks work as an encoding function of user history, which encodes user's underlying preference into compact vectors. Conventionally, user representations are well generated in the offline stage, making it independent of specific candidate item and incapable of making adaption for the recommendation context. Because of its simplicity, the candidate-independent representation is widely adopted in industry. However, without referencing to specific recommendation context, the candidate-independent representation will probably be too generalized to reflect user's preference from the interested aspect, which inevitably impairs the recommendation performance.

Recently, more advanced approaches are inspired by the popularity of attention mechanisms [Zhou *et al.*2018b, Zhou *et al.*2018a, Wang *et al.*2018]. By attentively aggregating user history w.r.t. the given candidate, user preference can be better captured from the perspective of interest. However, such kind of user representation has to be conducted at runtime, (i.e., everytime a specific candidate needs to be recommended for the user,) and it requires the access to the user's entire history. As a result, it will repetitively incur a series of time-consuming data retrieval operations. (For example, the access to a user's viewed ads involves the selection of her previous ads-clicks from activity logs, followed by the join with tables of ads' contents.) Considering that recommendations must be made in real-time for many applications, like news feed and online advertising, the deployment feasibility of such methods will be severely limited. In fact, it remains a challenging problem to have user represented with both superior recommendation efficacy and deployment feasibility.

To address this challenge, a novel user representation framework, High-Fidelity Archive Network (Hi-Fi Ark) is proposed, which makes the best of both worlds. Instead of allocating each user with one unique representation, Hi-Fi Ark compresses user history into a small constant number of "archives" in the offline stage (each of which is a vector). With this mild augmentation in space and judicious information extraction, it is expected that user preference can be comprehensively preserved with the highly compact archives. While deployed in the online service, these archives will be attentively aggregated w.r.t. the candidate item such that user preference can be fully captured from the interested perspective. To realize the above ambition, technical designs are carried out in the following ways.

First of all, one of the major deficiencies about the candidate-independent representation is that user history is summarized in a fixed and monotonous fashion, thus incapable of reflecting user preference comprehensively. In Hi-Fi Ark, the multi-head attentive pooling mechanism is employed, which encodes user history into multiple complementary vectors. Each of these vectors is deployed to reflect user's behavioral patterns from a specific aspect; there-

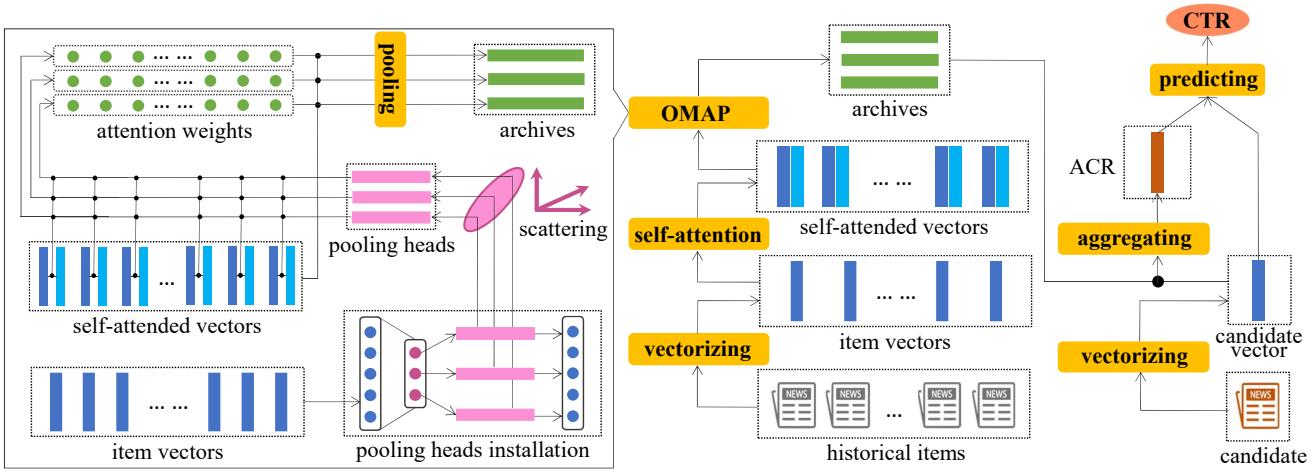

Figure 1: Schematic Illustration for Hi-Fi Ark's Architecture (better viewed in colour).

fore, user preference can be comprehensively preserved by such compact archives. However, the direct enforcement of multi-head attentive pooling may result in incompatibility and redundancy between different heads, which will severely limit their joint representation capability. Thus in the second place, the coordinated mechanism, namely orthogonal multi-head attentive pooling, is developed. On the one hand, the pooling heads are initially set as the orthogonal basis of the items' vectorization space. It is demonstrated that such a deployment of pooling heads is desirable, as the principle of user's behavioral patterns can be fully captured with it. On the other hand, to eliminate the redundancy from subsequent operations, an auxiliary regularization function is devised, which guarantees the pooling heads' complementarity all the way through the training process. Last but not least, given that user's historical items may not be precisely represented due to insufficient feature or inaccurate encoder, considerable noise might be introduced when representations are summarized from user history. To address this problem, the intra-correlation about user history is explicitly leveraged via self-attention. By aggregating each item's representation with those relevant in semantics, essences of user behaviors can be extracted more robustly.

Major contributions of this work are summarized into the following three points.

- A novel user representation framework, Hi-Fi Ark, is proposed in this work. By judiciously summarizing user history into compact archives, and attentively aggregating these archives w.r.t. the presented candidate, user's specific preference can be precisely captured without expensive access to the entire history.
- With the joint employment of orthogonal multi-head attentive pooling and self-attention, user preference can be captured comprehensively and robustly by the archives.
- Extensive empirical studies are performed with three real-world datasets on news recommendation, online advertising, and e-commerce, respectively. It is demonstrated that Hi-Fi Ark outperforms the well-recognized baseline methods remarkably and consistently, thereby verifying the effectiveness of our proposed methods.

The rest of this paper is organized as follows. The overview of Hi-Fi Ark and its detailed analysis are presented in Section 2. Then empirical studies of Hi-Fi Ark are conducted in Section 3, followed by the related work discussed in Section 4. Finally, conclusion of this work is made in Section 5.

2 Hi-Fi Archive Network

In this section, we will get started with the preliminaries and architecture of Hi-Fi Ark; then detailed analysis will be made for each of its components.

2.1 Preliminaries and Architecture

Hi-Fi Ark works with four basic steps: item vectorization, archive generation, archives' aggregation w.r.t. candidate and CTR prediction, whose pipeline is illustrated as Figure 1. First of all, user's historical items (e.g., news clicks) $\mathbf{V}^u$ are mapped from raw features into continuous vectors $\mathbf{\Theta}^u$, where items' semantics are encoded[1]. Secondly, the vectorized items $\mathbf{\Theta}^u$ are compressed into a small set of vectors $\mathbf{B}^u$, called archives. These archives serve as flexible candidate-independent representations for the user in the offline stage. Thirdly, given a candidate item $\acute{v}$ for recommendation, the archives are attentively aggregated w.r.t. the candidate's encoded vector $\theta_{\acute{v}}$, where user representation relevant to the candidate, $\phi^u_{\acute{v}}$, is produced. Finally, $\phi^u_{\acute{v}}$ and $\theta_{\acute{v}}$ are jointly processed by a specific feed forward function (e.g., multi-layer perception), where u's click through rate about $\acute{v}$ is obtained.

To distinguish from the original way of candidate-relevant user representation, which works directly with user's entire history, definitions for the Original Candidate-relevant Representation (OCR) and the Archived Candidate-relevant Representation (ACR) are introduced as follows.

Definition 1 *(OCR, ACR) Given vectorized items of user's entire history $\mathbf{\Theta}^u$, archives $\mathbf{B}^u$, and a candidate item $\acute{v}$, OCR, denoted as $\hat{\phi}^u_{\acute{v}}$, is generated as the attentive aggregation of u's entire history $\mathbf{\Theta}^u$ w.r.t. $\acute{v}$; meanwhile ACR, denoted as $\phi^u_{\acute{v}}$, is the attentive aggregation of the archives $\mathbf{B}^u$ w.r.t. $\acute{v}$.*

[1]Mapping functions are individualized for different data, concrete implementations are introduced in Section 3.

It seems that OCR will always be a more precise reflection for the user's preference towards $\acute{v}$, as it is made on top of user's entire history. However, with judicious extraction of user's informative behavioral patterns via Hi-Fi Ark, comparable performance can be achieved by the resulted ACR.

2.2 Orthogonal Multi-head Attentive Pooling

Each user's vectorized items are to be summarized with k archives through Orthogonal Multi-head Attentive Pooling (OMAP). In this place, a set of k vectors $\mathbf{\Lambda} = \{\lambda_1, ..., \lambda_k\}$, namely pooling heads, are introduced. With each head λ_i, user u's vectorized items are attentively aggregated into the archive β_i^u via the following operations:

$$\beta_i^u = \sum\nolimits_{\mathbf{\Theta}^u} \alpha_\theta^i \theta_v, \text{ where } \alpha_i^v = \frac{\exp\{\theta_v^T \lambda_i\}}{\sum_{\mathbf{\Theta}^u} \exp\{\theta_v^T \lambda_i\}}. \quad (1)$$

It is apparent that each pooling head summarizes user history from a unique perspective, and an increasing number of pooling heads tend to have a more comprehensive view of the user. However, it is not a plausible choice to make use of multiple heads casually. On the one hand, because of the underlying redundancy and incompatibility, it is unlikely for a small number pooling heads to get user represented comprehensively; on the other hand, a huge number of pooling heads, despite a more comprehensive view, require much more space to keep the archives, making it expensive and awkward for online deployment. To have users comprehensively represented in a compact way, the following mechanisms are developed on top of the fundamental multi-head attentive pooling.

Pooling Heads Installation. As is discussed, the efficacy of multi-head attentive pooling is greatly limited by its underlying redundancy and incompatibility. To overcome such a problem, the pooling heads are deployed w.r.t. the following pair of intuitions.

- The pooling heads should be distinguished from each other so that user history can be summarized from different perspectives; i.e., for each pair of $\lambda_i, \lambda_j \in \mathbf{\Lambda}$, it is desired that the similarity between them $sim(\lambda_i, \lambda_j)$ can be minimized.
- The pooling heads have to be deployed w.r.t. the vector distribution of the whole items $\hat{\mathbf{\Theta}}$ [2] (i.e., the items ever consumed by the users), such that the principle of user information can be fully preserved.

With both considerations, the pooling heads are initialized as the orthogonal basis of $\hat{\mathbf{\Theta}}$'s vector space. Particularly, such pooling heads can be acquired through the following optimization problem,

$$\min_{\mathbf{W}} \sum\nolimits_{\theta_v \in \hat{\mathbf{\Theta}}} \|\theta_v - \theta_v \mathbf{W}\mathbf{W}^T\|_2, \quad (2)$$

where θ_v is a d-dimension vector, and $\mathbf{W}$ is a $d \times k$ matrix. It is straightforward that the above problem is convex and fully differentiable, where the optimal solution $\mathbf{W}^*$ can be derived via gradient descent. Finally, each pooling head λ_i is set to be the i-th column of $\mathbf{W}^*$:

$$\lambda_i \leftarrow \mathbf{W}_i^*, \ 0 < i \leq k. \quad (3)$$

With the above way of pooling heads installation, the resulted archives can be comparable to the original entire history, as user preference from arbitrary aspect can be well preserved. Such a property is stated as Theorem 1.

[2] $\hat{\mathbf{\Theta}}$ can be easily acquired beforehand thanks to the popularity of unsupervised representation learning and pre-trained models.

Algorithm 1: End-to-End CTR Prediction

Input: user u's historical items $\mathbf{V}^u$, candidate item $\acute{v}$;
Output: click through rate $\omega(u, \acute{v})$;
begin
- Vectorize historical items $\mathbf{V}^u$ into $\mathbf{\Theta}^u$;
- Self-attention over $\mathbf{\Theta}^u$ for $\dot{\mathbf{\Theta}}^u$ via Eq. 7;
- Represent items as $\hat{\mathbf{\Theta}}^u \leftarrow \mathbf{\Theta}^u + \dot{\mathbf{\Theta}}^u$;
- Pooling heads installation for $\mathbf{\Lambda}$ via Eq. 2;
- Get user archives $\mathbf{B}^u$ via Eq. 1, on top of $\mathbf{\Lambda}$ and $\hat{\mathbf{\Theta}}^u$;
- Aggregate $\mathbf{B}^u$ w.r.t. $\hat{\theta}_{\acute{v}}$ and get ACR $\phi_{\acute{v}}^u$ via Eq. 4;
- Predict CTR $\omega(u, \acute{v})$ on top of $\phi_{\acute{v}}^u$ and $\hat{\theta}_{\acute{v}}$ via Eq. 8;

Theorem 1 *Archives $\mathbf{B}^u$ is a near lossless compression of user history $\mathbf{\Theta}^u$, as the transformations between ACR $\phi_{\acute{v}}^u$ and OCR $\hat{\phi}_{\acute{v}}^u$ are approximately invertible.*

Proof 1 *ACR and OCR, according to their definitions in Def. 1, are calculated as the attentive aggregations (w.r.t. $\acute{v}$) of user's archives and the entire history, respectively:*

$$\phi_{\acute{v}}^u = \sum\nolimits_{\mathbf{B}^u} \alpha_i \beta_i^u, \text{ where } \alpha_i = \frac{\exp\{\beta_i^{uT} \theta_{\acute{v}}\}}{\sum_{\mathbf{B}^u} \exp\{\beta_j^{uT} \theta_{\acute{v}}\}}, \quad (4)$$

$$\hat{\phi}_{\acute{v}}^u = \sum\nolimits_{\mathbf{\Theta}^u} o_v \theta_v, \text{ where } o_v = \frac{\exp\{\theta_v^T \theta_{\acute{v}}\}}{\sum_{\mathbf{\Theta}^u} \exp\{\theta_v^T \theta_{\acute{v}}\}}. \quad (5)$$

Provide that $\mathbf{W}^*$ is an orthogonal basis of $\hat{\mathbf{\Theta}}$, and products between vectors ($\theta_v^T \theta_{\acute{v}}$, $\theta_v^T \lambda_i$, $\beta_i^{uT} \theta_{\acute{v}}$) are small enough, the following relationships can be derived:

$$\begin{aligned}
\phi_{\acute{v}}^u &\propto (\mathbf{B}^u)^T \mathbf{B}^u \theta_{\acute{v}} + \sum\nolimits_{\mathbf{B}^u} \beta_i^u \\
&\propto ((\mathbf{\Theta}^u \mathbf{W}^*)^T \mathbf{\Theta}^u)^T (\mathbf{\Theta}^u \mathbf{W}^*)^T \mathbf{\Theta}^u \theta_{\acute{v}} + \sum\nolimits_{\mathbf{B}^u} \beta_i^u \\
&= (\mathbf{\Theta}^u)^T \mathbf{\Theta}^u (\mathbf{\Theta}^u)^T \mathbf{\Theta}^u \theta_{\acute{v}} + \sum\nolimits_{\mathbf{B}^u} \beta_i^u, \\
\hat{\phi}_{\acute{v}}^u &\propto (\mathbf{\Theta}^u)^T \mathbf{\Theta}^u \theta_{\acute{v}} + \sum\nolimits_{\mathbf{\Theta}^u} \theta_v.
\end{aligned}$$

(Notations are kind of abused here, as column vectors sets $\mathbf{B}^u$ and $\mathbf{\Theta}^u$ can also be treated as matrices.) Based on the above relationships, it is easy to verify that there will always exist a full rank $k \times k$ matrix $\mathbf{\Psi}$, which satisfies:

$$\begin{aligned}
(\mathbf{B}^u)^T \mathbf{B}^u &= \mathbf{W}^* \mathbf{\Psi} \mathbf{\Psi} (\mathbf{W}^*)^T, \\
(\mathbf{\Theta}^u)^T \mathbf{\Theta}^u &= \mathbf{W}^* \mathbf{\Psi} (\mathbf{W}^*)^T.
\end{aligned}$$

As a result, the following relationships can be derived:

$$\begin{aligned}
\phi_{\acute{v}}^u &\propto \mathbf{W}^* \mathbf{\Psi} (\mathbf{W}^*)^T (\hat{\phi}_{\acute{v}}^u - \hat{\varepsilon} \sum\nolimits_{\mathbf{\Theta}^u} \theta_v) + \hat{\varepsilon} \sum\nolimits_{\mathbf{B}^u} \beta_i^u, \\
\hat{\phi}_{\acute{v}}^u &\propto \mathbf{W}^* \mathbf{\Psi}^{-1} (\mathbf{W}^*)^T (\phi_{\acute{v}}^u - \varepsilon \sum\nolimits_{\mathbf{B}^u} \beta_i^u) + \varepsilon \sum\nolimits_{\mathbf{\Theta}^u} \theta_v.
\end{aligned}$$

therefore justifying the statement of Theorem 1. Notice that although dot-attention is used throughout the paper for better empirical performances, it's straightforward that the same conclusion holds for many other ways of attention, e.g., generalized dot $\mathbf{x}\mathbf{W}\mathbf{y}^T$ *and concatenation* $\mathbf{W}concat(\mathbf{x},\mathbf{y})^T$.

The above property indicates that user's informative behavioral patterns are fully preserved on top of the proposed pooling heads installation, based on which the preference toward the candidate can be extracted with high-fidelity.

Orthogonal Pooling Heads Scattering. Following the installation operation, the pooling heads will be further incorporated into an end-to-end working pipeline (to be discussed in Section 2.4), which is trained to predict the click through rate. To ensure that redundancy is excluded from the pooling heads during such a process, function $\mathbf{\Omega}$ is employed to restrict the changes of $\mathbf{\Lambda}$:

$$\mathbf{\Omega} = \|\mathbf{\Lambda}^T\mathbf{\Lambda} \circ (\mathbf{J_k} - \mathbf{I_k})\|_F, \tag{6}$$

In the above equation, $\mathbf{J_k}$ and $\mathbf{I_k}$ are the k-dimension all-one and identity matrices, respectively, and $\circ$ denotes the Hadamard product. $\mathbf{\Omega}$ is employed as a regularizer and restricted to be tiny during the training process. As a result, orthogonality is maintained as the inner products between different pooling heads are forced to be zeros.

2.3 Self-Attentive Item Vectorization

It is desired that historical items are precisely vectorized from their raw features, so that user preference can be truthfully extracted on top of them. However, this condition doesn't always hold in reality, given that insufficient features or inaccurate encoders will introduce tremendous noise to the vectorized items. To address this problem, self-attention is employed by Hi-Fi Ark, where the intra-correlation about user history is leveraged. Instead of working with the vectorized items directly, the self-attended vector $\dot{\theta}_v$ ([Vaswani *et al.*2017]) is calculated:

$$\dot{\theta}_v = \sum\nolimits_{\mathbf{\Theta}^u} \alpha_{v'}\theta_{v'}, \text{ where } \alpha_{v'} = \frac{\exp\{\theta_{v'}^T\theta_v\}}{\sum_{\mathbf{\Theta}^u}\exp\{\theta_{v'}^T\theta_v\}}. \tag{7}$$

As a result, each item will be aggregated with those relevant in semantics. It can be interpreted as the importance sampling for item v's underlying category, where the importance weights equal to the corresponding attention scores; with the aggregation of such samples, noise is resisted for each item and its true category can be better reflected. Following the common practice, the original vectorized item and self-attended vector are combined into the joint vector $\hat{\theta}_t$ via residual connection, i.e., $\hat{\theta}_v \leftarrow \dot{\theta}_v + \theta_v$.

2.4 End-to-End CTR Prediction

Once user's archives are well generated based on historical items, they will be deployed for the recommendation service. Presented with a candidate item $\acute{v}$, the archives will be attentively aggregated into the archived candidate-relevant representation $\phi_{\acute{v}}^u$ w.r.t. $\acute{v}$ (as indicated by Eq. 4), based on which user preference is precisely reflected from the perspective of interest. Both user and item's representations, $\phi_{\acute{v}}^u$ and $\theta_{\acute{v}}$, will be jointly processed by a certain type of feed forward network f so as to predict the click through rate $\omega(u,\acute{v})$:

$$\omega(u,\acute{v}) = f(\phi_{\acute{v}}^u, \theta_{\acute{v}}). \tag{8}$$

Finally, the unified end-to-end CTR prediction is shown as Alg. 1, based on which the whole model is trained to optimize the following binary classification problem,

$$\min \mathcal{F}_{\{\mathbf{U},\acute{\mathbf{V}}\}}(\omega(u,\acute{v}), \mathcal{I}(u,\acute{v})) + \tau\mathbf{\Omega}, \tag{9}$$

In Eq. 9, $\mathcal{F}$ calculates the binary cross entropy, $\mathcal{I}(u,\acute{v})$ represents the binary labels, and $\mathbf{\Omega}$ is the orthogonality regularizer defined in Eq. 6. Positive instances in $\{\mathbf{U},\acute{\mathbf{V}}\}$ are collected from the ground-truth clicks, while the negative ones are randomly sampled.

	News	Ads	E-Commerce
#users	10000	2956	15163
#items	49751	16375	45779
#clicks/u	84.42	10.74	52.39
feature	Text	DSSM	ID

Table 1: Dataset Statistics.

3 Empirical Studies

3.1 Experiment Settings

The following three datasets on news recommendation, online advertising and e-commerce are adopted for evaluation, whose statistics are shown in Table 1.

News. A total of 5-week news clicks are provided by MSN News from the EN-US market[3]. The dataset includes: 1) click relationships between users and news, 2) news titles, 3) news categories (e.g, sports, finance). Samples within the first four weeks are used for training, while those in the last week are used for testing. Titles are used as raw features, with 1D CNN employed as text encoder. The text encoder is pretrained for the classification of news category.

Ads. A total of one-week Ads clicks are offered by Bing Ads [Parsana *et al.*2018], which includes: 1) click relationships between users and URLs, 2) titles of URLs. The titles have already been mapped into 300-dimension vectors with a well pretrained DSSM model [Huang *et al.*2013] over massive corpus.

E-Commerce. This dataset[4] contains users' shopping behaviors on Amazon (the ratings-only dataset). All the purchased items are treated as positive cases, while negative cases are randomly sampled from the non-purchased ones. In contrast to other dataset, items in this dataset are purely represented by an unique ID, which is to be vectorized via a cold-started embedding matrix. As a result, we are able to evaluate the boundary case where recommendations have to be made with highly limited features.

The following baseline methods are jointly evaluated in our empirical studies.

[3]https://www.msn.com/en-us/news

[4]http://jmcauley.ucsd.edu/data/amazon/

	News			Ads			E-Commerce		
	AUC	NDCG@5	NDCG@10	AUC	NDCG@5	NDCG@10	AUC	NDCG@5	NDCG@10
AVG	0.6508/0.6535	0.3525/0.3546	0.4298/0.4318	0.8096/0.8108	0.5261/0.5307	0.5770/0.5817	0.7531/0.7647	0.4348/0.4582	0.5011/0.5161
CNN	0.6463/0.6481	0.3469/0.3488	0.4253/0.4267	0.8148/0.8207	0.5374/0.5478	0.5923/0.6032	0.6751/0.6806	0.358/0.3659	0.4214/0.4271
MUL	0.6567/0.6588	0.3555/0.3583	0.4333/0.4354	0.8593/0.8618	0.6464/0.6480	0.6845/0.6902	0.8421/0.8461	0.6151/0.6244	0.6543/0.6653
SNG	0.6469/0.6485	0.3505/0.3526	0.4275/0.4294	0.8426/0.8446	0.6041/0.6070	0.6470/0.6532	0.7741/0.7766	0.4746/0.4757	0.5320/0.5367
MEM	0.6423/0.6430	0.3457/0.3462	0.4242/0.4245	0.8448/0.8482	0.6471/0.6477	0.6827/0.6865	0.7712/0.7868	0.4689/0.4839	0.5303/0.5445
ATT	0.6476/0.6496	0.3507/0.3536	0.4282/0.4308	**0.8850/0.8866**	**0.7315/0.7364**	**0.7547/0.7585**	0.7066/0.7125	0.3983/0.4084	0.4572/0.4687
ARK-DE-3	0.6560/0.6565	0.3546/0.3550	0.4323/0.4328	0.8475/0.8513	0.6338/0.6427	0.6719/0.6802	0.8381/0.8433	0.5861/0.6012	0.6326/0.6453
ARK-EN-3	**0.6618/0.6631**	**0.3611/0.3635**	**0.4378/0.4395**	0.8779/0.8786	0.6711/0.6732	0.7083/0.7149	**0.8446/0.8535**	**0.6225/0.6396**	**0.6607/0.6750**

Table 2: Major experimental results (for every entry, the average performance is presented in the front, with the peak value followed in the second place; the highest scores across different methods are highlighted in bold).

Average Pooling (AVG). User representation ϕ_v^u is generated with the average pooling of user's historical vectorized items $\mathbf{\Theta}^u$.

1D CNN (CNN). User representation ϕ_v^u is generated with a 1D convolutional neural network followed by the average pooling operation over $\mathbf{\Theta}^u$.

Multi-head Attention (MUL). User archives are generated with multi-head attentive pooling at first as Hi-Fi Ark, but integrated into one unified vector with a $kd \times d$ mapping matrix, which follows [Vaswani *et al.*2017].

Single-head Attentive Pooling (SNG). A special case of multi-head attentive pooling, which only one single head is employed; thus, only one share of archive is generated.

KV Memory Network (MEM). External vectors known as keys are introduced, with which user history is summarized (written) and attentively aggregated (read) [Miller *et al.*2016].

Direct Attention (ATT). User representation is generated by attending the entire history $\mathbf{\Theta}^u$ w.r.t. the candidate, which is the mechanism in [Zhou *et al.*2018b, Wang *et al.*2018].

Because of limited space, Hi-Fi Ark is mainly evaluated for two variations with 3 pooling heads: the basic version (ARK-DE), with only multi-head attentive pooling activated, but OMAP and self-attention disabled; and the full version (ARK-EN), where all modules are enabled. Meanwhile, effects of different number of heads and each functional component are also studied. Notice that our method is for non-sequential data, thus those for temporal recommendation, e.g., Time LSTM [Zhu *et al.*2017], are not within discussion. We focus on evaluating the effectiveness of user representation, where a simple dot operation is used for CTR calculation (no obvious difference in performance with other simple alternatives, like MLP). However, sophisticated models, e.g., DeepFM, can be seamlessly combined. Finally, We evaluate the performance in terms of AUC scores and NDCG@(5, 10).

3.2 Experiment Analysis

The major experimental results are demonstrated in Table 2, where different methods are jointly compared over all three datasets. It can be clearly observed that ARK-EN-3 outperforms other methods on News and E-Commerce; meanwhile, it is merely second to ATT on Ads by a comparatively small margin. In our experiment, ATT is the only method, which makes recommendation on top of user's entire history (the others are based on summarization of user history). Intuitively, it should have made the best performances in all circumstances. However, we have a somewhat unexpected observation: as ARK-EN-3 beats ATT on both News and E-Commerce, moreover, ARK-EN-3's performance is way better than ATT on E-Commerce. Much of this phenomenon is resulted from the differences in feature granularity. Particularly, ATT heavily relies on fine-grained features, as it will attend every single item in history to identify the relevant context to a query; ARK-*, however, is less sensitive to the feature granularity of individual items, as it works with summarization of all items in user history. Both Ads and News datasets make uses of items' texts as their raw features, but texts of Ads are encoded with a powerful DSSM model [Huang *et al.*2013] well-trained over massive corpus, which characterizes items much more precisely. On the other hand, only item IDs are available in the E-Commerce, where no explicit semantic information is incorporated, which leads to much weaker characterization for the items in contrast to the others. Such properties align exactly with their demonstrated performances in the experiment, which confirms our analysis. At the same time, we can also observe that ARK-EN-3 is consistently better than ARK-DE-3, which indicates that OMAP and self-attention are necessary for improving the efficacy of multi-head attentive pooling. A more detailed analysis is to be made in the next part.

Effects of different components and number of heads are studied with the News dataset. Results on other datasets are omitted due to similar observations and space limit.

Componential Analysis. Componential effect is studied (in Table 3-A), with every single component, pooling heads installation (-IN), orthogonal pooling heads scattering (-OR) and self-attention (-SL), added on top of the multi-head attentive pooling. Meanwhile, the basic (-DE) and full (-EN) versions are also included for comparison. It can be observed that all the components are playing positive roles, as consistent improvements are achieved over the basic ARK-DE-3.

Hi-Fi Ark's Implementation is available at https://github.com/xyyimian/Hifi-Ark/

A. Component Effect			
	AUC	NDCG@5	NDCG@10
ARK-DE-3	0.6560/0.6565	0.3546/0.3550	0.4323/0.4328
ARK-IN-3	0.6567/**0.6601**	0.3550/0.3575	0.4329/0.4351
ARK-OR-3	**0.6588**/0.6597	**0.3583/0.3595**	**0.4353/0.4361**
ARK-SL-3	**0.6588**/0.6595	0.3571/0.3592	0.4345/0.4360
ARK-EN-3	**0.6618/0.6631**	**0.3611/0.3635**	**0.4378/0.4395**
B. #Heads Effect			
	AUC	NDCG@5	NDCG@10
SNG	0.6469/0.6485	0.3505/0.3526	0.4275/0.4294
ARK-EN-3	**0.6618**/0.6631	**0.3611/0.3635**	**0.4378/0.4395**
ARK-EN-5	0.6604/0.6610	0.3579/0.3594	0.4354/0.4364
ARK-EN-10	0.6607/**0.6638**	0.3592/0.3628	0.4360/0.4394

Table 3: Component (A) and #Heads (B) Effects (the highest average and peak values are marked in bold black; the 2nd highest values in A are marked in bold blue).

Moreover, the combination of all components, i.e., ARK-EN-3, outperforms all the other baselines. Such a phenomenon is within our expectation, that different components work from complementary perspectives, thus magnifying the effect of every single component when integrated as a whole.

Effect of #heads. Effect of different number heads is studied with ARK-EN-*, where the pooling heads are set to be 3, 5 and 10, respectively. Besides, the special case of single-head attentive pooling (SNG) is adopted for comparison. According to the results in Table 3-B, the employment of multiple heads does bring a remarkable leap forward, as significant improvements are achieved over SNG. However, having even more heads will not continuously benefit the efficacy, as ARK-EN-5&10 show almost no increment over ARK-EN-3. Recall that the adoption of multiple heads is to ensure the comprehensive summarization of user history; and it is found in our experiment that a rank-3 setting is already capable of reconstructing all the vectorized items in the News dataset with ignorable loss (i.e., the solution of Eq. 2). Thus, there's no wonder to see the efficacy of heaving merely 3 heads. Actually, the reconstruct loss can be used as an important reference for the selection of heads number in practice.

4 Related Work

In this section, relate work is analyzed from two aspects: deep learning based recommendation and deep user representation.

Deep Learning-based Recommendation. Deep learning-based recommendation systems have become popular in recent years. Thanks to its powerful feature representation capability, deep learning techniques were firstly employed in recommendation system so as to introduce diverse sources of information; e.g., convolutional neural networks were applied for taking advantage of visual information [He and McAuley2016], denoised auto encoder [Zhang *et al.*2016] and recurrent neural networks [Bansal *et al.*2016] were adopted for exploiting textual information, and network embedding [Zhang *et al.*2016] was used to incorporate relational information. Meanwhile, deep learning is used to learn combinatorial features [Cheng *et al.*2016, Guo *et al.*2017, Lian *et al.*2018, Wang *et al.*2017a], where meaningful feature combinations are selected as discriminative patterns via deep neural networks. Finally, deep learning is also adapted as alternatives to other classical algorithms, such as matrix factorization [He *et al.*2017, Xue *et al.*2017].

Deep User Representation. Analogous to the representation learning of visual and textual information, deep learning is widely used to model user behavior, whose compact and informative representation will greatly benefit the efficacy of downstream recommendation. Roughly speaking, user representation can be generated in two ways: the candidate-independent approach and the candidate-relevant approach. Particularly, user representation can be generated in the offline stage purely based on user history [Elkahky *et al.*2015, Covington *et al.*2016], which will be directly deployed for online recommendation services. Such kind of user representation is independent of specific candidate item, and it is widely adopted in real-world industry owing to its simplicity. On the other hand, user representation may also be generated at runtime [Zhou *et al.*2018b, Zhou *et al.*2018a, Wang *et al.*2018], by attentively aggregating user history w.r.t. specific candidate. It is found that better recommendation performances can be achieved with the attentive aggregation of user history, where similar conclusions have also been reported in other areas like Question Answering and Semantic Matching [Wang *et al.*2017b, Santos *et al.*2016, Lin *et al.*2017]. Unfortunately, it is not always possible to meet the requirement of runtime access to users' entire history, making it inapplicable for many real-world scenarios. Recent works [Zhang *et al.*2017, Chen *et al.*2018] also make use of memory networks [Sukhbaatar *et al.*2015, Miller *et al.*2016] to model user's behaviors. The memory networks are structurally close to Hi-Fi Ark, as multiple memory units are introduced for the attentive aggregation of specific queries. However, little effort is made previously on preserving user's comprehensive preference with compact vectors. As is theoretically analyzed and empirically validated in our work, simple augmentation in space is far from enough for such a purpose.

5 Conclusion

A novel user representation framework, Hi-Fi Ark, is proposed in this work. With the joint employment of orthogonal multi-head attentive pooling and self-attention, user history is comprehensively summarized into compact archives; and by attentively referencing to each candidate item, user preference can be precisely captured from the perspective of interest. Thus, superior recommendation efficacy comes along with strong feasibility in Hi-Fi Ark. Extensive experiments are conducted over various real-world datasets, whose results demonstrate the effectiveness of our proposed approaches.

Acknowledgements

The authors would like to thank Microsoft News for providing technical support and data in the experiments, and Jiun-Hung Chen and Ying Qiao for their support and discussions.

References

[Bansal *et al.*, 2016] Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In *RecSys*, pages 107–114, 2016.

[Chen *et al.*, 2018] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential recommendation with user memory networks. In *WSDM*, pages 108–116, 2018.

[Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[Covington *et al.*, 2016] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys*, pages 191–198, 2016.

[Elkahky *et al.*, 2015] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*, pages 278–288, 2015.

[Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[He and McAuley, 2016] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *AAAI*, pages 144–150, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338, 2013.

[Lian *et al.*, 2018] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *KDD*, pages 1754–1763, 2018.

[Lin *et al.*, 2017] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[Miller *et al.*, 2016] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

[Parsana *et al.*, 2018] Mehul Parsana, Krishna Poola, Yajun Wang, and Zhiguang Wang. Improving native ads ctr prediction by large scale event embedding and recurrent networks. *arXiv preprint arXiv:1804.09133*, 2018.

[Santos *et al.*, 2016] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*, 2016.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2017a] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *ADKDD*, page 12, 2017.

[Wang *et al.*, 2017b] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, volume 1, pages 189–198, 2017.

[Wang *et al.*, 2018] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *WWW*, pages 1835–1844, 2018.

[Xue *et al.*, 2017] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In *IJCAI*, pages 3203–3209, 2017.

[Zhang *et al.*, 2016] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *KDD*, pages 353–362, 2016.

[Zhang *et al.*, 2017] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *WWW*, pages 765–774, 2017.

[Zhou *et al.*, 2018a] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. *arXiv preprint arXiv:1809.03672*, 2018.

[Zhou *et al.*, 2018b] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *KDD*, pages 1059–1068, 2018.

[Zhu *et al.*, 2017] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *IJCAI*, pages 3602–3608, 2017.