Retrieval-Oriented Knowledge for Click-Through Rate Prediction

Huanshuo Liu
National University of Singapore
Singapore
huanshuo.liu@u.nus.edu

Bo Chen
Huawei Noah's Ark Lab
China
chenbo116@huawei.com

Menghui Zhu
Huawei Noah's Ark Lab
China
zhumenghui1@huawei.com

Jianghao Lin
Shanghai Jiao Tong University
China
chiangel@sjtu.edu.cn

Jiarui Qin
Shanghai Jiao Tong University
China
qinjr@icloud.com

Yang Yang
Huawei Noah's Ark Lab
China
yangyang590@huawei.com

Hao Zhang
Huawei Noah's Ark Lab
Singapore
zhang.hao3@huawei.com

Ruiming Tang ✉
Huawei Noah's Ark Lab
China
tangruiming@huawei.com

Abstract

Click-through rate (CTR) prediction plays an important role in personalized recommendations. Recently, sample-level retrieval-based models (e.g., RIM) have achieved remarkable performance by retrieving and aggregating relevant samples. However, their inefficiency at the inference stage makes them impractical for industrial applications. To overcome this issue, this paper proposes a universal plug-and-play retrieval-oriented knowledge (**ROK**) framework. Specifically, a knowledge base, consisting of a retrieval-oriented embedding layer and a knowledge encoder, is designed to preserve and imitate the retrieved & aggregated representations in a decomposition-reconstruction paradigm. Knowledge distillation and contrastive learning methods are utilized to optimize the knowledge base, and the learned retrieval-enhanced representations can be integrated with arbitrary CTR models in both instance-wise and feature-wise manners. Extensive experiments on three large-scale datasets show that ROK achieves competitive performance with the retrieval-based CTR models while reserving superior inference efficiency and model compatibility.

1 Introduction

Click-through rate (CTR) prediction is a key component of many personalized online services, such as recommender systems [37] and web search [8, 10, 20]. It seeks to estimate the probability of a user's click given a particular context [41]. The CTR models could be mainly classified into two categories. The first category is the feature interaction-based methods. The core idea of these methods is to capture the high-order feature interactions across multiple fields with different operators (*e.g.,* product [3, 12, 26, 35], convolution [21, 39], and attention [30, 38]). The second category is the user behavior modeling methods that leverage different architectures (*e.g.,* RNN [13, 14], CNN [32], attention [43, 44], memory bank [22, 27]) to extract informative knowledge from user behavior sequences for final CTR prediction.

To further enhance the performance, UBR4CTR [25] and SIM [23] retrieve useful behaviors from the user's behavior history (*i.e.,* clicked items), reducing the potential noise in user behavior sequences. Subsequent studies [19] have improved these methods' efficiency using hashing functions [5] and parallel retrieval execution [1] during inference. Recent studies [9, 24] further expand these retrieval-based methods from **item-level** retrieval to **sample-level** retrieval, applicable to general CTR prediction settings. Instead of retrieving similar items from the **user history**, RIM [24] adopts the idea of k nearest neighbor (kNN) and designs a sample-centric retrieval method, which aggregates the relevant data samples retrieved from the **search pool** (*e.g.,* the whole training dataset). PET [9] constructs a hypergraph over the retrieved data instances and performs message propagation to get a better representation of the target data for final CTR prediction.

Although sample-level retrieval-based methods bring impressive performance enhancement, they have to perform instance-wise comparisons between the target data sample and each candidate sample in the search pool (usually million or even billion level). This leads to extreme inefficiency problems during inference, making it impractical for industrial applications, as illustrated in the left part of Figure 1. Recent work, such as DERT [42], has improved the retrieval mechanisms by using vector retrieval techniques, enhancing efficiency. However, DERT's dependency on the RIM Encoder and its $O(\log N)$ retrieval time complexity, along with the additional cost of encoding numerous samples in the retrieval pool, limits its scalability and effectiveness in large-scale applications.

This work introduces the Retrieval-Oriented Knowledge (ROK) framework to tackle the inference inefficiency problem of sample-level retrieval-based methods, with two stages: Retrieval-Oriented Knowledge Construction and Knowledge Utilization. Initially, we pre-train a sample-level retrieval-based model, such as RIM[24]. To resolve the inefficient inference issue, ROK tries to learn a neural network-based Knowledge Base to imitate the aggregated representations from the pre-trained retrieval-based methods via a Retrieval Imitation module, whose simple diagram is shown in Figure 1. The Knowledge Base, built on a novel **decomposition-reconstruction paradigm**, where a Retrieval-Oriented Embedding Layer captures the *feature-wise* embedding and a Knowledge Encoder reconstructs the *instance-wise* aggregated representations (*i.e.,* retrieval-enhanced representation). In this way, instead of the time-consuming retrieval, efficient inference of neural networks could be adopted to get the approximated aggregated representations. Additionally, we introduce a Contrastive Regularization module to ensure learning stability and prevent model collapse. At the Knowledge Utilization stage, ROK designs two approaches (i.e., instance-wise and feature-wise) to integrate the retrieval-enhanced representations with various CTR models, thus providing high inference efficiency.

The main contributions of this paper are as follows:

- In this work, we introduce ROK, a pioneering framework that transforms sample-level retrieval-based methods —previously deemed infeasible in the industry into a practical solution. Our approach innovatively leverages a neural network-based Knowledge Base to efficiently store the retrieval-enhance representations and hence obviates the need for time-intensive retrieval & aggregation during inference.
- We use knowledge distillation and contrastive learning to optimizie the knowledge base, allowing the integration of retrieval-enhanced representations with various CTR models, both at the instance and feature levels.
- Extensive experiments conducted across three large-scale datasets have shown that ROK not only achieves competitive performance when compared to existing retrieval-based CTR methods but also reserves superior inference efficiency. Moreover, ROK enhances the performance of various CTR methods due to its exceptional compatibility. This success underscores the effectiveness of the neural knowledge model as a compact surrogate for the retrieval pool.

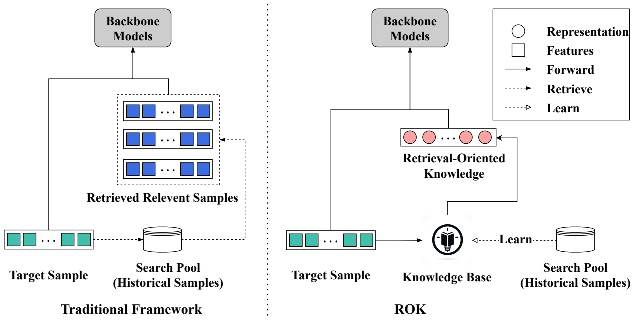


Figure 1: Comparison between traditional sample-level retrieval framework and ROK.

2 Related Work

2.1 CTR Prediction

For the click-through rate (CTR) prediction, models can essentially be divided into two main categories: those focusing on feature interaction and those centering on user behavior modeling. The first category is the feature interaction-based methods, evolving from foundational works such as POLY2 [2] and Factorization Machines (FM) [28]. With the integration of Deep Neural Networks (DNNs), a variety of sophisticated deep feature interaction models have been proposed. These models aim to capture high-order feature interactions across different fields by employing various operations, such as the product operation[3, 12, 18, 26, 35], convolution [21, 39], and attention mechanisms [30, 38]. The key innovation of these models lies in their ability to identify and utilize complex interactions between a multitude of features to enhance the predictive performance of CTR models.

Besides, user behavior modeling is another core technique for CTR prediction that mines user preferences from historical interaction behaviors meticulously. To better extract informative knowledge from user's behavior sequence, various network structures have been utilized, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Attention Networks, and Memory Networks. GRU4Rec [14] designs Gated Recurrent Units (GRUs) to capture the preference-evolving relationship while Caser [32] leverages the horizontal and vertical convolution to model skip behaviors at both union-level and point-level. Moreover, the attention mechanism is the most popular method for modeling item dependencies, and several influential works have been proposed, including SASRec [17], DIN [44], DIEN [43], and BERT4Rec [31]. Among these, DIN and DIEN leverage the target-attention network to identify important historical items, while SASRec and BERT4Rec apply the self-attention network with Transformer architecture to excavate behavior dependencies. Besides, memory-based methods [15] are also proposed to store user behavior representations in a read-write manner.

2.2 Retrieval-Augmented Recommendation

To further enhance the performance of CTR prediction, the retrieval-augmented recommendation is proposed, where the most relevant information is retrieved from the historical data. Specifically, UBR4CTR [25] and SIM [27] are designed to retrieve beneficial behaviors from the user's historical extremely long behavior sequence, where UBR4CTR deploys the search engine method while SIM uses the hard search and soft search approaches. To make the search procedure end-to-end, ETA [6] is proposed by leveraging the SimHash algorithm to map the user behavior into a low-dimensional space, hence achieving learnable retrieval. Moreover, recent works further extend the retrieval-augmented recommendation from **item-level** retrieval to **sample-level** retrieval by retrieving informative samples. RIM [24] is the first to deploy this method that leverages the search engine to retrieve several relevant samples from the search pool and performs neighbor aggregation. PET [9] and DERT [42] have respectively made improvements in the interaction and retrieval mechanism of neighboring samples. PET constructs a hypergraph over the retrieved data samples and performs message propagation to improve the target data representations for final CTR prediction. DERT utilizes vector retrieval to speed up the retrieval process.

3 Preliminary

3.1 CTR Prediction

In CTR prediction, each data sample is denoted as $s_t = (x_t, y_t)$, where $x_t = \{c_i^t\}_{i=1}^F$, F is the number of discrete features[1], and y_t is the label. Thus a dataset with N samples can be expressed as $T = \{s_t\}_{t=1}^N$. The goal of the CTR prediction is to estimate the click probability of a specific sample: $\hat{y}_t = G(x_t; \theta)$, where G is the CTR model with learnable parameters θ.

[1] Continuous features are usually discretized with various methods [11].

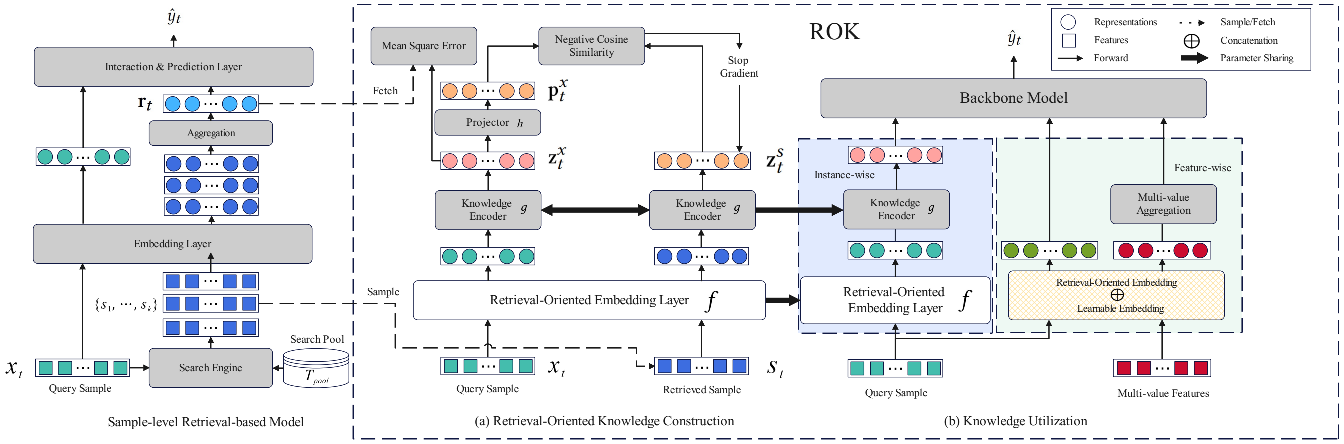


Figure 2: Overall framework of ROK. The process begins with the Retrieval-Oriented Knowledge Construction stage, where a sample-level retrieval-based model has already been pre-trained, and the retrieval and aggregation results are offline pre-stored. A tailored knowledge base is then designed to imitate the aggregation results, with Contrastive Regularization ensuring stabilization during the training phase. This design allows the time-consuming retrieval process to be substituted with the swift forward propagation of a neural network. In the subsequent Knowledge Utilization stage, the framework introduces the retrieval-enhanced representation into backbone CTR models in both an instance-wise and feature-wise approach, ensuring the maintenance of sample-level retrieval abilities while circumventing the complexities of retrieval and aggregation operations.

Apart from the traditional training/validation/testing splits for CTR prediction, retrieval-based methods further require a search pool T_{pool}, which might overlap with the training set T_{train} according to different settings. The search pool T_{pool} is constructed to provide useful knowledge for downstream CTR prediction towards the target sample x_t, which can be formulated as:

$$\hat{y}_t = G\left(x_t, R\left(x_t\right); \theta\right), \tag{1}$$

where $R\left(x_t\right)$ is the retrieved knowledge of x_t. For ***item-level*** retrieval [23, 25], the knowledge $R\left(x_t\right)$ is the retrieved k user behaviors. For ***sample-level*** retrieval [9, 24], the knowledge $R\left(x_t\right)$ is the retrieved k nearest data samples. As for our proposed ***ROK***, the knowledge $R\left(x_t\right)$ is the learned retrieval-enhanced representation.

After obtaining the click prediction $\hat{y}_t$, the parameters θ are optimized by minimizing the binary cross-entropy (BCE) loss:

$$\mathcal{L}_t = y_t \log \hat{y}_t + (1 - y_t) \log(1 - \hat{y}_t). \tag{2}$$

3.2 Sample-level Retrieval-based Methods

For illustration purposes, we abstract the sample-level retrieval approach in the left part of Figure 2. The target sample x_t will be considered as a query to retrieve the top-K neighboring samples $\{s_1, \cdots, s_K\}$ from the search pool T_{pool} through the search engine. After retrieval, an aggregation layer is used to aggregate the features and labels of the retrieved samples, resulting in a dense knowledge representation $\mathbf{r}_t$. For example, RIM [24] deploys an attentive aggregation layer, while PET [9] constructs a hypergraph over the retrieved samples, and performs message propagation to aggregate the representations. Finally, the aggregated representation $\mathbf{r}_t$ is sent to the prediction module together with the target sample representation. The prediction module usually contains components for feature interaction modeling [3, 12, 30] or user behavior modeling [43, 44].

4 Methodology

4.1 Overview of ROK

Despite the superior performance, sample-level retrieval-based methods suffer from the inference inefficiency problem due to the time-consuming online retrieval process. To this end, we propose a novel Retrieval-Oriented Knowledge (ROK) framework, where a knowledge base is built to imitate the aggregated retrieval knowledge $\mathbf{r}_t$. As shown in Figure 2, ROK consists of two stages: (1) Retrieval-Oriented Knowledge Construction, and (2) Knowledge Utilization.

In the first stage of ROK–*retrieval-oriented knowledge construction*–we pre-train a sample-level retrieval-based model[2], followed by designing a delicate *knowledge base* with a retrieval-oriented embedding layer and a knowledge encoder. This is designed to imitate the retrieval & aggregation results (*i.e.,* $\mathbf{r}_t$) of the pre-trained model by directly generating the final aggregated representation $\mathbf{z}_t^x$ through the knowledge encoder. In this way, when online serving, the time-consuming retrieval process could be replaced by a simpler and faster forward propagation of a neural network (*i.e.,* knowledge base). We adopt the mean square error loss for knowledge imitation and design a contrastive regularization loss to stabilize the learning process and prevent collapse [16, 29].

In the *knowledge utilization* stage, we propose to integrate the plug-and-play retrieval-enhanced representations into arbitrary backbone CTR models in both *instance-wise* and *feature-wise* manner. Hence, we retain the sample-level retrieval capacity, while avoiding the online overhead brought by retrieval & aggregation operations.

Next, we will first elaborate on the network structure design of the **knowledge base** for knowledge imitation. Then, we give the

[2] In this work, we employed RIM[24] which is one of the foundational works in sample-level retrieval-based methods. The pre-training process exactly follows the original work. Importantly, our proposed method extends its applicability to encompass all sample-level retrieval-based methods.

detailed training strategies for the two stages (*i.e.,* retrieval-oriented knowledge construction and knowledge utilization).

4.2 Structure Design of Knowledge Base

For sample-level retrieval-based methods [9, 24], the overhead of inference inefficiency is mainly caused by the retrieval & aggregation operations, which are uncacheable and unavoidable during the online inference if a brand-new data sample comes. To this end, we propose a novel **decomposition-reconstruction paradigm**, where the aggregated representation $\mathbf{r}_t$ is first decomposed into feature-level embeddings, and then reconstructed via a neural encoder. As illustrated in 3, our proposed knowledge base comprises two key modules: (1) retrieval-oriented embedding layer f, and (2) knowledge encoder g.

Initially, we decompose the query x_t into *feature-wise* embeddings via the retrieval-oriented embedding layer f. Then, the obtained feature-wise embeddings are fed into the knowledge encoder g to reconstruct the *instance-wise* aggregated representation $\mathbf{z}_t^x = g(f(x_t))$. Specifically, a learnable embedding layer functions as the retrieval-oriented embedding layer. The architecture of knowledge encoder g can be chosen arbitrarily (*e.g.,* transformer [34]). For simplicity, we adopt multi-layer perceptron (MLP) in this paper. This approach reduces the time complexity of sample-level retrieval from $O(N \log(N))$, where N is the search pool size to $O(1)$. Since we introduce the retrieval-oriented embedding layer f, we will cut down the embedding size of f from d to $d/2$ for fair comparison in space complexity. With the help of the learned knowledge base, online retrieval & aggregation operations can be avoided.

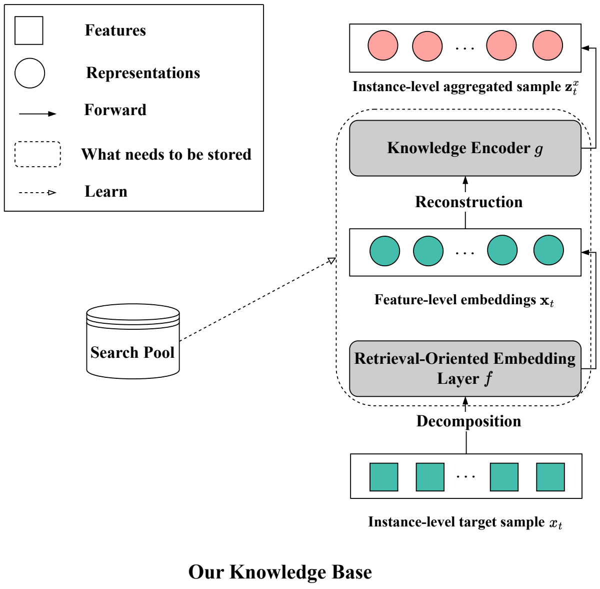

Features
Representations
Forward
What needs to be stored
Learn
Instance-level aggregated sample $\mathbf{z}_t^x$
Knowledge Encoder g
Reconstruction
Feature-level embeddings $\mathbf{x}_t$
Search Pool
Retrieval-Oriented Embedding Layer f
Decomposition
Instance-level target sample x_t
Our Knowledge Base

Figure 3: Structure Design of Knowledge Base.

4.3 Retrieval-Oriented Knowledge Construction

To construct the knowledge base, we introduce two modules: Retrieval Imitation and Contrastive Regularization.

4.3.1 Retrieval Imitation After we obtain the reconstructed representation $\mathbf{z}_t^x$ from the knowledge base for query sample x_t. We use mean square error (MSE) loss to train the knowledge base, enabling it to imitate the aggregated representation $\mathbf{r}_t$:

$$\mathcal{L}_{imit} = \mathrm{MSE}\left(\mathbf{z}_t^x, \mathbf{r}_t\right). \tag{3}$$

This knowledge imitation approach serves as a knowledge distillation process, where we extract and inject the retrieval-enhanced knowledge from the pre-trained model into the learnable knowledge base, enabling the implicit sample-level retrieval capability.

4.3.2 Contrastive Regularization Besides, we have intricately integrated a contrastive regularization loss to ensure stability in the learning process and avert potential collapse [16, 29]. Notably, sample-level retrieval-based methods [9, 24] ingeniously produces positive samples during the retrieval process. This can be perceived as a unique data augmentation technique, offering a localized positive perspective for contrastive learning. Consequently, ROK employs the SimSiam [7] framework, utilizing a free negative samples scheme for regularization. While the traditional sample-level retrieval-based methods predominantly harness global features from the aggregated neighboring samples, our use of contrastive regularization allows for a more detailed extraction of local features from these samples.

As shown in Figure 2(a), the most correlated neighboring sample s_t is selected from the K retrieved samples $\{s_1, \cdots, s_K\}$ by the sample-level retrieval-based methods. Then, the target sample x_t and selected neighboring sample s_t are fed into the knowledge base to obtain the reconstructed representations $\mathbf{z}_t^x$ and $\mathbf{z}_t^s$, respectively. To prevent collapse in the absence of negative samples [40], a projector h is employed to generate the projected representation $\mathbf{p}_t^x$, $\mathbf{p}_t^s$ from $\mathbf{z}_t^x$, $\mathbf{z}_t^s$.

$$\mathbf{p}_t^x \triangleq h\left(\mathbf{z}_t^x\right) \triangleq h\left(g\left(f\left(x_t\right)\right)\right), \tag{4}$$

$$\mathbf{p}_t^s \triangleq h\left(\mathbf{z}_t^s\right) \triangleq h\left(g\left(f\left(s_t\right)\right)\right). \tag{5}$$

The cosine similarity between projected representation $\mathbf{p}_t^x$ and reconstructed representations $\mathbf{z}_t^s$ is defined as:

$$\mathcal{D}\left(\mathbf{p}_t^x, \mathbf{z}_t^s\right) = \frac{\mathbf{p}_t^x}{\left\|\mathbf{p}_t^x\right\|_2} \cdot \frac{\mathbf{z}_t^s}{\left\|\mathbf{z}_t^s\right\|_2}, \tag{6}$$

where $\|\cdot\|_2$ is the l_2 norm. Besides, the gradient of $\mathbf{z}_t^s$ is stopped when computing the loss [7, 40]. To avoid spatial bias, motivated by the Jensen–Shannon (JS) divergence, a symmetric loss is adopted and the final contrastive regularization loss is:

$$\mathcal{L}_{contra} = -\left(\frac{1}{2}\mathcal{D}\left(\mathbf{p}_t^x, \mathbf{z}_t^s\right) + \frac{1}{2}\mathcal{D}\left(\mathbf{p}_t^s, \mathbf{z}_t^x\right)\right). \tag{7}$$

Hence, The overall objective for ROK in the *retrieval-oriented knowledge construction* stage is:

$$\mathcal{L} = (1-\alpha) \cdot \mathcal{L}_{imit} + \alpha \cdot \mathcal{L}_{contra}, \tag{8}$$

where α adjusts the ratio of the two loss terms. After the retrieval-oriented knowledge construction stage, the knowledge base will be retained for the follow-up knowledge utilization stage.

4.4 Knowledge Utilization

In the knowledge utilization stage, the plug-and-play retrieval-enhanced representations from the learned knowledge base can be integrated into various backbone CTR models. When integrated with backbone models, all parameters of ROK are frozen—a strategy that outperforms others, as detailed in Section 5.5.1. As illustrated in Figure 2(b), we present two approaches for knowledge utilization, which are mutually compatible to boost the final CTR prediction:

- **Feature-wise retrieval-enhanced** method leverages the retrieval-oriented embedding layer f to obtain the retrieval-enhanced feature-wise embeddings for the target sample. Specifically, this method integrates the retrieval-oriented embedding layer f with the original learnable feature embedding of the backbone model to form the final embedding layer. [3] .
- **Instance-wise retrieval-enhanced** method employs the knowledge base to generate a retrieval-enhanced instance-level aggregated representation $\mathbf{z}_t^x$ for each sample. Typically, various backbone models produce their unique representations—for example, RIM's aggregated representations and DIEN's interest states. These unique representations are then concatenated with the feature embeddings and the combined results are fed into the interaction and MLP (Multi-Layer Perceptron) layers.

$$\begin{cases} \mathbf{z}_t^x &= g(f(x_t)) \\ \hat{y}_t &= \text{backbone}(\phi(x_t) \oplus \mathbf{z}_t^x) \end{cases},$$

where ϕ is utilized to extract the unique representations of the backbone model.

5 Experiments

In this section, we present the experimental settings and corresponding results in detail. The experiments are conducted on three large-scale datasets, including Tmall, Taobao, and Alipay. To gain more insights into ROK, we endeavor to address the following research questions (RQs) in this section.

- **RQ1:** How does ROK's performance compare to that of traditional CTR models and retrieval-based methods?
- **RQ2:** How compatible is ROK with other backbone models?
- **RQ3:** How does the knowledge captured by ROK contribute to improving performance?
- **RQ4:** How does the chosen update strategy and positive sample selecting strategy affect ROK's performance?
- **RQ5:** How do the specified hyperparameter influence ROK's performance?

5.1 Experimental Settings

5.1.1 Datasets. The evaluations were conducted on three widely-recognized public datasets: Tmall, Taobao, and Alipay. Comprehensive statistics for these datasets are presented in Table 1. Specifically, we tallied the counts of users, items, samples, fields, and categories for each dataset. It's noteworthy to mention that the number of features refers to the count of unique feature values.

Table 1: The dataset statistics.

Dataset	Users #	Items #	Samples #	Fields #	Features #
Tmall	424,170	1,090,390	54,925,331	9	1,529,676
Taobao	987,994	4,162,024	100,150,807	4	5,159,462
Alipay	498,308	2,200,291	35,179,371	6	3,327,205

Following RIM [24], we organized the data such that the oldest instances constituted the search pool, the most recent instances comprised the test set, and those in between were assigned to the training set. The retrieval-based methods [24, 25] retrieve neighboring samples from the search pool. For user behavior modeling methods [43, 44] involved (e.g., DIEN), the sequential features (e.g., user behavior) are also generated from the search pool.

5.1.2 Evaluation Metrics. To measure the performance, we utilize the commonly adopted metrics AUC and log-loss, which reflect pairwise ranking performance and point-wise likelihood, respectively. A significance test contrasting the two top-performing methods for each metric is undertaken, with results of significance denoted by an asterisk ($*$).

We follow [44] to introduce Rel. Impr. metric to quantify the relative improvement of models, which is defined as below:

$$\text{Rel. Impr.} = \left(\frac{\text{AUC(measured model)} - 0.5}{\text{AUC(base model)} - 0.5} - 1 \right) \times 100\%, \quad (9)$$

where the base model is the backbone model of ROK in each dataset.

5.1.3 Compared Baselines. For baselines, we compare ROK with a mid-tier performance backbone model against traditional models and retrieval-based models. For traditional CTR models, GBDT [4] is a frequently utilized tree-based model, and DeepFM [12] is a popular deep-learning model that focuses on feature interactions. HPMN [27] and MIMN [22] utilize memory network architectures, while DIN [44] and DIEN [43] are attention-based CTR models designed to pinpoint user interests. Additionally, FATE [36] is a model for learning representations of tabular data which facilitates interactions among samples within a minibatch. In item-level retrieval-based models, SIM [23] and UBR [25] extract pertinent user behaviors from a comprehensive set of user-generated data. As for the sample-level retrieval-based model, RIM [24] retrieves relevant data instances based on raw input features, and DERT [42] retrieves dense representations. To ensure a fair comparison, we adopted the same hyper-parameter settings for the baselines and the ROK backbone during the knowledge utilization phase as those used in prior work [24].

Regarding backbone models, we intentionally select average-performing common models like DIN and DIEN rather than top-tier ones like UBR, which will further improve the performance of ROK. By default, we choose DIEN [43] for Tmall and Taobao datasets and DIN [44] for the Alipay dataset. The substantial improvement observed in 5.2 when these backbone models are combined with ROK underscores the efficacy of ROK.

5.2 Overall Performance Comparison: RQ1

In this section, we compare the overall experimental results in Table 2, from which we have the following observations. First, ROK remarkably boosts the backbone model with AUC improved by 10.08%, 34.96%, and 16.85% on the three datasets respectively, demonstrating the superior performance. Second, retrieval-based methods [24, 25] significantly outperform user behavior modeling methods [43, 44], benefiting from their superior knowledge retrieval ability. Moreover, sample-level retrieval-based methods [24] perform better than item-level retrieval-based methods [23, 25] due to more comprehensive retrieval. Finally, ROK markedly outperforms the item-level retrieval-based methods such as SIM [23] and UBR [25], and achieves performance surpassing that of the sample-level retrieval-based methods like RIM [24]. Although ROK is only

[3] The embedding layer accounts for a large proportion of the total number of parameters in CTR models. For a fair comparison in model parameters, the embedding size of both the retrieval-oriented embedding layer and the original learnable feature embedding of the backbone model is set to $d/2$ compared with other models.

Table 2: Performance comparison of CTR prediction task with various baselines. For each dataset, RIM is selected as the teacher model, and an average-performing backbone model is selected for ROK: DIEN for both the Tmall and Taobao datasets and DIN for the Alipay dataset. $^{\circ}$ indicates recent results. Evaluation metrics include AUC and log-loss (LL). Among the models excluding recent work, the best results are highlighted in bold, while the runner-ups are underlined. The gray-colored bold text indicates recent results outperform all other models. "Rel. Impr." signifies the model's relative AUC improvement over the chosen backbone model, with a statistical significance level of $p < 0.01$.

Category	Model	Tmall			Taobao			Alipay		
		AUC	LL	Rel. Impr	AUC	LL	Rel. Impr	AUC	LL	Rel. Impr
Traditional Models	GBDT	0.8319	0.5103	-13.55%	0.6134	0.6797	-54.75%	0.6747	0.9062	-34.00%
	DeepFM	0.8581	0.4695	-6.72%	0.671	0.6497	-31.76%	0.6971	0.6271	-25.54%
	FATE	0.8553	0.4737	-7.45%	0.6762	0.6497	-29.69%	0.7356	0.6199	-10.99%
	HPMN	0.8526	0.4976	-8.15%	0.7599	0.5911	3.71%	0.7681	0.5976	1.28%
	MIMN	0.8457	0.5008	-9.95%	0.7533	0.6002	1.08%	0.7667	0.5998	0.76%
	DIN	0.8796	0.4292	-1.12%	0.7433	0.6086	-2.91%	0.7647	0.6044	0.00%
	DIEN	0.8839	0.4272	0.00%	0.7506	0.6082	0.00%	0.7485	0.6019	-6.12%
Retrieval-based Models	SIM (Item-level)	0.8857	0.4520	0.47%	0.7825	0.5795	12.73%	0.7600	0.6089	-1.78%
	UBR (Item-level)	0.8975	0.4368	3.54%	0.8169	0.5432	26.46%	0.7952	0.5747	11.52%
	RIM (Teacher model)	0.9151	0.3697	8.13%	**0.8567***	**0.4546***	42.34%	0.8005	0.5736	13.52%
	DERT$^{\circ}$ (Sample-level)	0.9200	0.3585	9.40%	**0.8647**	**0.4486**	45.53%	0.8087	0.5319	16.62%
Our Model	ROK	**0.9226***	**0.3546***	10.08%	0.8382	0.5098	34.96%	**0.8093***	**0.5304***	16.85%

based on knowledge distillation and contrastive learning, it not only surpasses the teacher model but also comes very close to recent results in terms of performance. Notably, in our methodology, we deliberately opted for models of average performance, such as DIN and DIEN, over elite models like UBR [25], with the anticipation that the latter would enhance the efficacy of ROK. Concurrently, it is pertinent to note that the teacher model employed within ROK is RIM, which DERT outperforms. Nonetheless, the adoption of a superior teacher model is poised to markedly augment the performance of ROK. Despite this, ROK has successfully enabled user behavior modeling methods to outperform both item-level retrieval-based methods and sample-level retrieval-based methods. This indicates that the neural knowledge model successfully serves as a compact surrogate for the retrieval pool. More importantly, the sample-level retrieval-based methods, due to their prolonged inference time, were once deemed entirely infeasible for industrial application, which will further be elaborated in Section 6.2. However, with the introduction of ROK, this barrier has been overcome. ROK not only mitigates the challenges but revolutionizes the deployment prospects, rendering sample-level retrieval-based methods feasible industry applications.

5.3 Compatibility Analysis: RQ2

In this section, we commence by contrasting the compatibility of our novel approach, ROK, with conventional CTR models, with the findings delineated in Table 3. Upon examining the data, two observations arise: (i) The application of ROK markedly amplifies the performance of the backbone models. To illustrate, by integrating ROK with DeepFM, DIN, and DIEN, there's a notable enhancement in performance metrics, with an average AUC increase of 4.30%, 8.01%, and 5.59%, coupled with an average decrease in log-loss by 11.80%, 8.51%, and 8.22% across three datasets, respectively. This highlights the significant benefits of incorporating the retrieval-oriented knowledge from ROK into the backbone models. (ii) Exhibiting model-agnosticism, ROK seamlessly integrates with a diverse array of backbone CTR models, irrespective of whether they focus on feature interaction or behavior modeling. This underscores that the knowledge garnered by ROK is beneficial across diverse models, highlighting the compatibility of ROK. The feature of model-agnosticism is crucial since, regardless of the optimal model for a given scenario, ROK consistently boosts its capabilities. Thus, instead of being bound to one model, we can choose the ideal one for each situation and leverage ROK to optimize its performance.

5.4 Comparison of the learned knowledge of ROK and RIM: RQ3

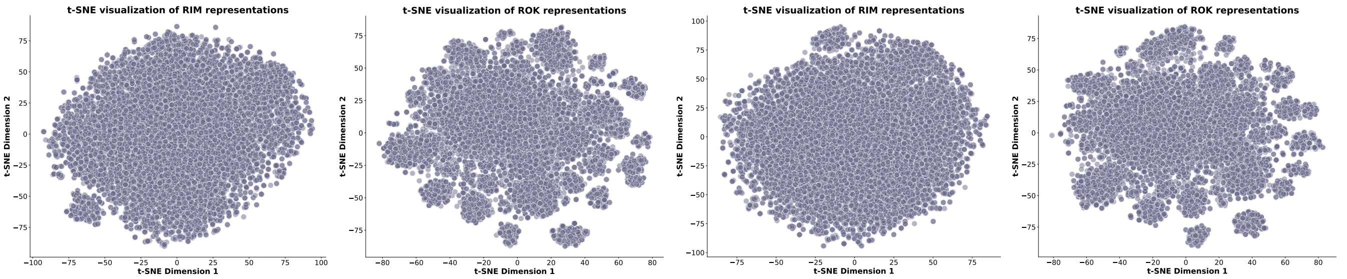
In this section, we delve into the comparative analysis of the knowledge quality acquired by ROK and RIM. For a detailed assessment, we formulated two simplest variants (LR and MLP) for each model to fully demonstrate the effect of knowledge:

- For ROK, we designed the ROK(LR) and ROK(MLP) variants. Both of these take as input the concatenation of the original feature embeddings $\mathbf{x}_t$, with ROK's knowledge component z_t^x.
- Correspondingly, for RIM, we introduced the RIM(LR) and RIM(MLP) variants. Their input comprises the concatenation of the original feature embeddings, $\mathbf{x}_t$, with RIM's aggregated features and labels, $\mathbf{r}_t$.

The detailed performance comparison of these variants is tabulated in Table 4 and we have two observations. First, significantly, the results indicate that the ROK variants surpass the RIM variants in performance across both datasets. This distinction in outcomes emphasizes the superior quality of knowledge learned by ROK, especially as these models apply this intrinsic knowledge directly for predictions. It's compelling to note that while ROK leverages the retrieval imitation strategy (detailed in Section 4.3.1) to imitate the aggregated features and labels of RIM, it extracts richer and more profound knowledge from the data. We are inclined to believe that the contrastive regularization methodology is pivotal in driving this

Table 3: Compatibility analysis of applying ROK to different backbone models. "Rel. Impr." means relative AUC and log-loss improvements of ROK against the backbone. Improvements are statistically significant with $p < 0.01$.

Model	Tmall				Taobao				Alipay			
	AUC	Rel. Impr.	LL	Rel. Impr.	AUC	Rel. Impr.	LL	Rel. Impr.	AUC	Rel. Impr.	LL	Rel. Impr.
DeepFM	0.8585	4.21%	0.4803	9.31%	0.6710	3.52%	0.6497	2.51%	0.6971	5.59%	0.6271	4.80%
DeepFM+ROK	**0.8946***	-	**0.4356***	-	**0.6946***	-	**0.6334***	-	**0.7361***	-	**0.5970***	-
DIN	0.8796	4.26%	0.4292	5.59%	0.7433	8.85%	0.6086	6.82%	0.7647	5.83%	0.6044	12.24%
DIN+ROK	**0.9171***	-	**0.4052***	-	**0.8091***	-	**0.5671***	-	**0.8093***	-	**0.5304***	-
DIEN	0.8839	4.38%	0.4272	20.50%	0.7506	11.67%	0.6082	16.18%	0.7485	5.33%	0.6019	7.61%
DIEN+ROK	**0.9226***	-	**0.3546***	-	**0.8382***	-	**0.5098***	-	**0.7884***	-	**0.5561***	-



(a) Alipay (Train) **(b) Alipay (Test)**

Figure 4: The t-SNE visualization of knowledge from RIM (on the left in each subfigure) and ROK (on the right in each subfigure) for Alipay.

enhanced performance for ROK. Second, **ROK(MLP)** in Tmall even outperforms DeepFM and some user behavior modeling methods, which shows another way to improve the pre-ranking model effect.

To delve deeper into the inherent knowledge of ROK and explain the superiority of ROK over RIM, we employed t-SNE visualization [33]. Figure 4 and Figure 7 depict the knowledge distribution patterns of both ROK and RIM across the Alipay and Tmall datasets. For this visualization, we randomly selected 10,000 samples from both the training and testing sets for each model. We have two observations. First, a discerning observation from Figure 4 reveals that ROK's knowledge distribution exhibits a more pronounced clustering effect across all datasets. This enhanced clustering can be attributed to contrastive regularization. For comparison, RIM's retrieved neighboring samples may include some incidental noise that correlates with the quantity of these samples. Besides, the aggregation mechanism of RIM only captures the global features of the neighboring samples. However, the contrastive regularization in ROK emphasizes the local features of neighboring samples and mitigates the noise of neighboring samples by selectively considering the most relevant sample, as shown in section 4.3.2. Specifically, representations of analogous samples are pulled closer, while those of dissimilar samples are pushed apart, leading to a more distinct and meaningful clustering. Secondly, a significant concern when using neural networks as a knowledge base in our case relates to their generalization abilities. This is because the training process is guided by the retrieved examples of sample-level retrieval-based methods, while the testing phase requires the model to apply its knowledge independently. Nonetheless, the visualization of ROK's representations reveals a high degree of similarity between the training and test datasets, showing the good generalization of the decomposition-reconstruction paradigm.

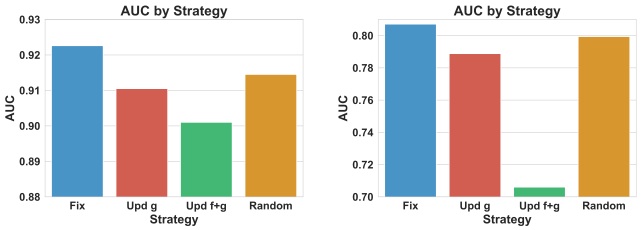
Table 4: Comparison of model variants. "Rel. Impr." means relative AUC improvement of ROK's variants against baselines and RIM's variants. Improvements are statistically significant with $p < 0.01$.

Model	Tmall		Alipay	
	AUC	Rel. Impr.	AUC	Rel. Impr.
LR	0.8213	4.37%	0.6298	8.96%
RIM(LR)	0.8379	2.30%	0.6613	3.77%
ROK(LR)	**0.8572***	-	**0.6862***	-
MLP	0.8393	2.98%	0.6344	9.13%
RIM(MLP)	0.8426	2.58%	0.6773	2.21%
ROK(MLP)	**0.8643***	-	**0.6923***	-

5.5 Ablation & Hyper-parameter Study

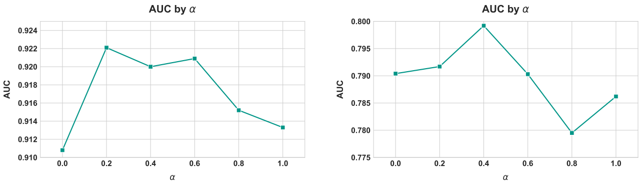
5.5.1 Update Strategy: RQ4.

In this section, we undertake comprehensive ablation experiments for ROK. We start by examining the parameter update strategy employed for the knowledge base during the knowledge utilization phase. The strategies under consideration include: fixing the knowledge base (*Fix*), updating only the upper knowledge encoder g (*Upd* g), and updating both the knowledge encoder and the retrieval-oriented embedding layer f (*Upd* $f + g$). Furthermore, we consider the positive sample selection strategy mentioned in Section 4.3.2. Specifically, we select the most related neighboring sample by default. Now, we compare the performance of randomly selecting a sample from the retrieved samples (*random*). Results obtained on the Tmall and Alipay datasets are



(a) Tmall (b) Alipay

Figure 5: Comparison of AUC scores for different update strategies.



(a) Tmall (b) Alipay

Figure 6: Comparison of AUC scores across varying α values.

depicted in Figure 5. We have two observations: First, we observe that randomly selecting a sample as the positive sample diminishes performance. This suggests that even within the narrowly retrieved set of samples, noise is still present. Second, a key observation from these results is that the strategy of fixing the knowledge base (*Fix*) yields the best performance. This can be attributed to the primary goal of ROK: to derive superior representations. However, updating with the backbone models may adversely impact the acquired knowledge. This characteristic is highly desirable and aligns well with our expectations, further enhancing the practicality of ROK. This indicates that once ROK has been fully trained, it can function as an independent module. This eliminates the need for subsequent modifications, reinforcing its role as an authentic knowledge base.

5.5.2 Hyperparameter Study: RQ5. In this section, we delve into the study of the hyperparameter α pertaining to the balance between the contrastive regularization loss and retrieval imitation loss, represented as α. As illustrated in Figure 6, for both datasets, when α assumes values of either 0 or 1—representing the extreme scenarios—the performance of ROK deteriorates. This observation underscores the significance of both retrieval imitation loss and contrastive regularization loss to the model's efficacy.

6 Industry Application

In this section, we will demonstrate the superior performance of ROK in training and inference efficiency.

6.1 Training Efficiency

In evaluating model efficiency, it's vital to look at both inference speed and training time. Training ROK involves pre-training a sample-level retrieval-based model, creating retrieval-oriented knowledge, and applying this knowledge. Despite these steps, the extra training time is reasonable compared to a standalone model. The training duration is detailed in Table 5. Experiments were run on an AMD EPYC 7T83 with an RTX 4090, using hyperparameters—learning rate, batch size, weight decay—optimized via grid search as per the RIM study. We also implemented early stopping after 3 steps to enhance efficiency. Two observations stand out. Firstly, incorporating ROK into the backbone model significantly reduces training time, thanks to fewer parameters needing updates, partly because ROK's parameters are frozen and the embedding size is halved. This leads to faster convergence. Second, the total time of Phase 1 and Phase 2 is close to the training duration of the DIEN without ROK. The relatively time-consuming part of the whole training process would be the RIM's retrieval mechanism. However, this is mitigated by using large bulk sizes and caching, which decreases retrieval times. For further efficiency, extending the knowledge base update frequency beyond the backbone model's could be beneficial.

Table 5: Training duration comparison for two phases, total time and backbone model on Tmall, Taobao, and Alipay, measured in minutes. Phase 1 includes data retrieval for training and testing tests, pre-training RIM, and Retrieval-Oriented Knowledge Construction. Phase 2 includes Knowledge Utilization, with DIEN as the backbone model. Additionally, DIEN training times without ROK integration are provided. For Phase 1, data retrieval times are specified separately.

Phase	Tmall	Taobao	Alipay
Phase 1 (Retrieval)	156 (40)	116 (30)	24 (4)
Phase 2	201	163	68
Total	357	279	92
DIEN without ROK	327	330	79

6.2 Inference Efficiency

In the real world of online services, from advertising platforms to recommendation systems, the significance of inference efficiency cannot be ignored. The introduction of the knowledge base significantly amplifies the efficiency in sample-level retrieval, trimming down the time complexity from $O(N \log(N))$, where N is the search pool size to a constant time $O(1)$ through bypassing conventional retrieval & aggregation mechanisms.

Within this context, we thoroughly assess the inference efficiency of various backbone models—DeepFM [12], DIEN [43], and DIN [44]—when seamlessly integrated with ROK and juxtaposed against retrieval-based methods such as UBR [25], RIM [24], and DERT [42]. The comparisons drawn from our analyses, as detailed in Table 6, offer insights. Specifically, due to the $O(N \log(N))$ complexity of the online retrieval process, the inference time of retrieval-based methods is significantly longer than other baseline models. For instance, RIM's inference time is an overwhelming $1 \sim 2$ orders of magnitude longer than DeepFM, DIEN, and DIN. This significant delay, despite any performance enhancements, fails to align with the real-time response expectations of RSs. These systems necessitate a swift response, making such latencies of retrieval-based methods industrially **infeasible**. Moreover, deploying retrieval-based methods for real-world recommendation systems remains complex and costly due to challenges such as encoding discretized features to dense representations, high resource consumption, and divergent inference processes. In contrast, our innovative ROK distinguishes itself by delivering notable improvements over the backbone models while maintaining negligible additional inference time, making sample-level retrieval-based methods **feasible in online service deployment**. As shown in Table 6, ROK consistently improves the AUC of backbone models across all datasets while only slightly increasing the inference speed by 0.12ms to 0.25ms per sample. Compared to retrieval-based methods, ROK achieves comparable or even better performance with significantly faster inference speed. For example, on the Tmall dataset, DIEN+ROK surpasses the performance of DERT (0.9226 vs. 0.9200) while being more than 3 times faster in inference (5.51ms vs. 17.78ms per sample).

ROK's design allows it to be retrieval-free and compatible with existing recommendation systems. It only needs to perform inference once, avoiding additional data transfer overhead and making it highly efficient. This represents a blend of predictive accuracy and rapid inference, making ROK a noteworthy solution for modern online services.

Table 6: Comparison of AUC and inference speed (ms per sample) across models on Tmall, Taobao, and Alipay.

Model	Tmall		Taobao		Alipay	
	AUC	Inference Speed	AUC	Inference Speed	AUC	Inference Speed
DeepFM	0.8585	1.34	0.6710	1.36	0.6971	1.19
DeepFM+ROK	0.8946	1.46	0.6946	1.40	0.7361	1.41
DIEN	0.8839	5.44	0.7506	4.89	0.7485	3.69
DIEN+ROK	0.9226	5.51	0.8382	5.04	0.7884	3.73
DIN	0.8796	1.28	0.7433	1.32	0.7647	1.43
DIN+ROK	0.9171	1.46	0.8091	1.36	0.8093	1.56
UBR (Retrieval)	0.8975	20.71 (17.67)	0.8169	56.45 (53.32)	0.7952	30.32 (27.31)
RIM (Retrieval)	0.9151	174.81 (173.27)	0.8567	206.22 (204.78)	0.8005	113.95 (112.37)
DERT (Retrieval)	0.9200	17.78 (16.19)	0.8647	19.53 (16.97)	0.8087	17.95 (16.31)

7 Conclusion

This paper introduces the ROK framework, addressing the inefficiency challenges inherent in sample-level retrieval-based methods during inference. By developing a neural network-based knowledge base, ROK imitates and preserves the essential retrieval representations through a novel decomposition-reconstruction approach, further optimized via knowledge distillation and contrastive learning. This allows for the integration of retrieval-oriented knowledge with various CTR models, either at an instance or feature level. Through rigorous testing on three public datasets, ROK not only matches the performance of existing retrieval-based CTR models but also demonstrates superior inference efficiency. These findings highlight ROK's potential to significantly improve the efficiency and effectiveness of CTR models.

References

[1] Yue Cao, Xiaojiang Zhou, Jiaqi Feng, Peihao Huang, Yao Xiao, Dayao Chen, and Sheng Chen. 2022. Sampling Is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2974–2983.

[2] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin. 2010. Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research* 11, 4 (2010).

[3] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. Enhancing explicit and implicit feature interactions via information sharing for parallel deep CTR models. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3757–3766.

[4] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. 1996. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering* 8, 6 (1996), 866–883.

[5] Qiwei Chen, Changhua Pei, Shanshan Lv, Chao Li, Junfeng Ge, and Wenwu Ou. 2021. End-to-end user behavior retrieval in click-through rateprediction model. *arXiv preprint arXiv:2108.04468* (2021).

[6] Qiwei Chen, Yue Xu, Changhua Pei, Shanshan Lv, Tao Zhuang, and Junfeng Ge. 2022. Efficient Long Sequential User Data Modeling for Click-Through Rate Prediction. arXiv:2209.12212 [cs.IR]

[7] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15750–15758.

[8] Xinyi Dai, Jianghao Lin, Weinan Zhang, Shuai Li, Weiwen Liu, Ruiming Tang, Xiuqiang He, Jianye Hao, Jun Wang, and Yong Yu. 2021. An adversarial imitation click model for information retrieval. In *Proceedings of the Web Conference 2021*. 1809–1820.

[9] Kounianhua Du, Weinan Zhang, Ruiwen Zhou, Yangkun Wang, Xilong Zhao, Jiarui Jin, Quan Gan, Zheng Zhang, and David Wipf. 2022. Learning enhanced representations for tabular data via neighborhood propagation. *arXiv preprint arXiv:2206.06587* (2022).

[10] Lingyue Fu, Jianghao Lin, Weiwen Liu, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. An F-shape Click Model for Information Retrieval on Multi-block Mobile Pages. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1057–1065.

[11] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2910–2918.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[13] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[15] Xunqiang Jiang, Binbin Hu, Yuan Fang, and Chuan Shi. 2020. Multiplex memory network for collaborative filtering. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 91–99.

[16] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348* (2021).

[17] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[18] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 2022. IntTower: the Next Generation of Two-Tower Model for Pre-Ranking System. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3292–3301.

[19] Xiaochen Li, Jian Liang, Xialong Liu, and Yu Zhang. 2022. Adversarial Filtering Modeling on Long-term User Behavior Sequences for Click-Through Rate Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1969–1973.

[20] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.

[21] Bin Liu, Ruiming Tang, Yingzhi Chen, Jinkai Yu, Huifeng Guo, and Yuzhou Zhang. 2019. Feature generation by convolutional neural network for click-through rate prediction. In *The World Wide Web Conference*. 1119–1129.

[22] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction.

In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[23] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[24] Jiarui Qin, Weinan Zhang, Rong Su, Zhirong Liu, Weiwen Liu, Ruiming Tang, Xiuqiang He, and Yong Yu. 2021. Retrieval & Interaction Machine for Tabular Data Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1379–1389.

[25] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. 2020. User behavior retrieval for click-through rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2347–2356.

[26] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)* 37, 1 (2018), 1–35.

[27] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, et al. 2019. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 565–574.

[28] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[29] Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. 2022. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*. PMLR, 19847–19878.

[30] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[33] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[35] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*. 1785–1797.

[36] Qitian Wu, Chenxiao Yang, and Junchi Yan. 2021. Towards open-world feature extrapolation: An inductive graph learning approach. *Advances in Neural Information Processing Systems* 34 (2021), 19435–19447.

[37] Yunjia Xi, Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Rui Zhang, Ruiming Tang, and Yong Yu. 2023. A Bird's-eye View of Reranking: from List Level to Page Level. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 1075–1083.

[38] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[39] Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon M Jose. 2019. CFM: Convolutional Factorization Machines for Context-Aware Recommendation.. In *IJCAI*, Vol. 19. 3926–3932.

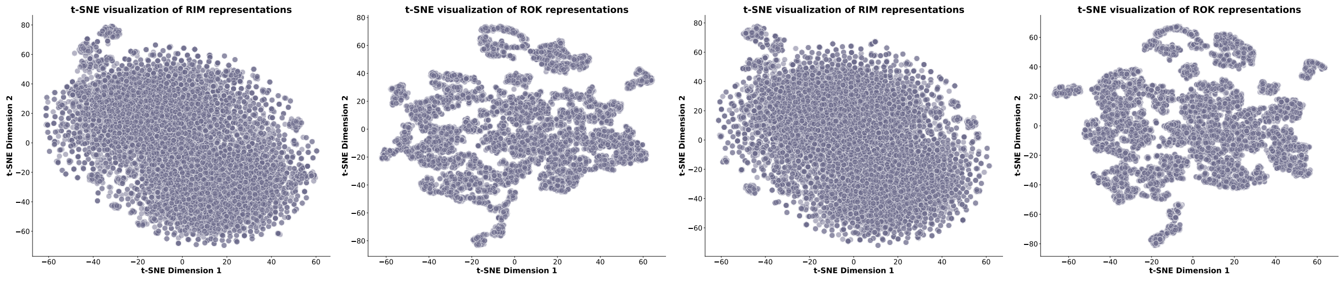
[40] Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X Pham, Chang D Yoo, and In So Kweon. 2022. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. *arXiv preprint arXiv:2203.16262* (2022).

[41] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584* (2021).

[42] Lei Zheng, Ning Li, Xianyu Chen, Quan Gan, and Weinan Zhang. 2023. Dense Representation Learning and Retrieval for Tabular Data Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3559–3569.

[43] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[44] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.



(a) Tmall (Train) **(b) Tmall (Test)**

Figure 7: The t-SNE visualization of knowledge from RIM (on the left in each subfigure) and ROK (on the right in each subfigure) for Tmall.

A Appendix

A.1 T-sne visualization

Figure 7 depict the knowledge distribution patterns of both ROK and RIM in Tmall dataset.

A.2 Ablation & Hyper-parameter Study on Taobao

In this section, we present the ablation and hyper-parameter study conducted on the Taobao dataset. The results are depicted in Figure 8, Figure 9.

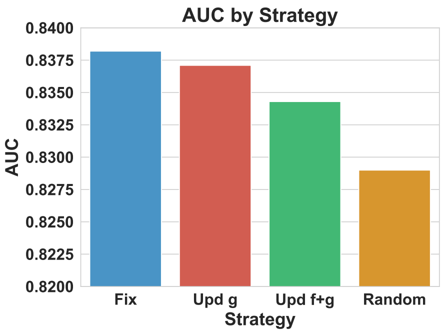


Figure 8: Comparison of AUC scores for different update strategies on Taobao.

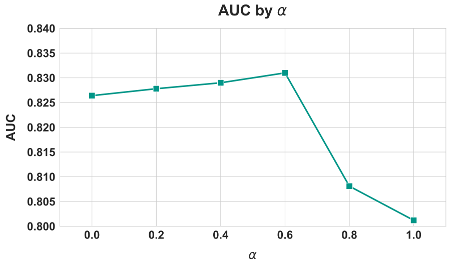


Figure 9: Comparison of AUC scores across varying α values on Taobao. At $\alpha = 0$, only Retrieval Imitation loss is utilized, whereas at $\alpha = 1$, only the Contrastive Regularization loss is applied.