Retrieval and Distill: A Temporal Data Shift-Free Paradigm for Online Recommendation System

Lei Zheng
zhenglei@roro-ai.com
RORO-AI PTE. LTD.
Singapore, Singapore

Ning Li
lining01@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Weinan Zhang*
wnzhang@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Yong Yu
yyu@apex.sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

Current recommendation systems are significantly affected by a serious issue of temporal data shift, which is the inconsistency between the distribution of historical data and that of online data. Most existing models focus on utilizing updated data, overlooking the transferable, temporal data shift-free information that can be learned from shifting data. We propose the Temporal Invariance of Association theorem, which suggests that given a fixed search space, the relationship between the data and the data in the search space keeps invariant over time. Leveraging this principle, we designed a retrieval-based recommendation system framework that can train a data shift-free relevance network using shifting data, significantly enhancing the predictive performance of the original model in the recommendation system. However, retrieval-based recommendation models face substantial inference time costs when deployed online. To address this, we further designed a distill framework that can distill information from the relevance network into a parameterized module using shifting data. The distilled model can be deployed online alongside the original model, with only a minimal increase in inference time. Extensive experiments on multiple real datasets demonstrate that our framework significantly improves the performance of the original model by utilizing shifting data.

KEYWORDS

Temporal Data Shift, Retrieval-Enhanced Methods, Knowledge Distillation, Pretraining, Click-Through Rate

1 INRODUCTION

Click-through rate (CTR) prediction models play a significant role in modern recommendation systems. To enhance the accuracy of CTR models, deep learning-based approaches have been extensively employed [3, 31, 36, 37]. However, the dynamic environment faced by the recommendation systems, with the data distribution evolving over time, poses significant challenges to the effectiveness of these models [50].

To tackle this challenge, prior research has undertaken investigations from two distinct perspectives, namely incremental learning and behavior sequence modeling. On one hand, incremental learning techniques delve into updating the models that make these predictions as new data comes in, without having to start from scratch every time [16, 21, 43]. This means the models learn and improve gradually as they get access to more recent data. However, the process of learning neural network parameters depends on iterative updates based on gradients, which slowly infuse supervision data into the model weights using a minimal learning rate. This poses a significant challenge for large parametric recommendation models in swiftly adapting to shifts in distribution. On the other hand, behavior sequence modeling focuses on understanding how users' behaviors change over time by looking at the sequence of actions they take. In this area, various network structures have been proposed, including recurrent neural networks (RNNs) [10, 11], convolutional neural networks (CNNs) [34, 44], and memory networks [23, 28]. Moreover, the attention mechanism has emerged as the predominant approach for modeling item dependencies, with several notable contributions such as SASRec [15], DIN [49], DIEN [48], and BERT4Rec [32]. These studies aim to identify sequential patterns in user behavior, thereby facilitating the prediction of future preferences based on historical activities. However, these studies fail to directly tackle the issue of distribution shift during training time, and these high-capacity models significantly increase the online workload and inference burden.

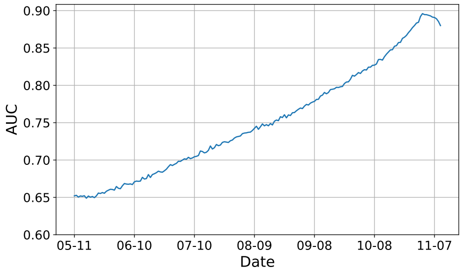

Figure 1: The phenomenon of temporal data shift on a real dataset collected from Tmall. The test set consists of data after Nov. 10, 2015. The x-axis represents the date from which we add all data up to Nov. 10 to train a CTR model, and the y-axis represents the model's performance on the test set. We found that the model's performance peaks after adding just a few final days of data. However, further adding data from earlier periods leads to a decline in the model's performence on the test set due to the inconsistency in data distribution between this earlier data and the test set.

[*] The corresponding author is Weinan Zhang

All of the above-mentioned models either update the model using the latest data [7, 36] or use past shifting data as a reference for retrieval [24, 26], without directly using the shifting data for training model parameters. However, directly using the temporally shifting data for model training is challenging, as an example illustrated in Figure 1 based on a public dataset of Tmall, an international e-commerce platform. In this dataset, data spanning Nov. 11-12, 2015 is employed as the test set. The x-axis represents the training set from the indicated date to Nov. 10, 2015. We train a common deep CTR model PNN [27] using different training sets to make predictions. The y-axis indicates the AUC performance of the model on the test set. We can observe that utilizing data over Nov. 1-10, 2015 for predictions on the test set yields the best results, whereas adding data from earlier time periods will negatively impact the results due to data shift. Therefore, it is crucial to find an effective method that can directly utilize these data for model training.

Inspired by the recent proposed retrieval-based tabular data prediction models [25, 47], we found that similar data from historical records can significantly enhance the predictive capability and generalizability. In other words, although the probability distribution modeled by the CTR is time-variant, the probability distribution of the association between current user behavior data and similar historical data can remain invariant over time if the search space is fixed. If we design a neural network to model the probability distribution of data similar to the current query, this neural network is temporal data shift-free. Thus, the shifting data can be used to train such a neural network. By incorporating this neural network with the original CTR model, we can obtain an aggregated prediction model that has been trained with shifting data.

In this paper, we propose the Retrieval and Distill paradigm (RAD) to address the challenge of ineffective training of a CTR model over shifting data. The RAD paradigm generally consists of two major components, namely the Retrieval Framework and the Distill Framework. We first establish a Retrieval Framework to assist the original model in making predictions using similar data within the shifting data. We refer to the neural network in Retrieval Framework that retrieves and outputs the logits of similar data as the relevance network. By pre-training this relevance network using shifting data and then merging the original CTR model and relevance network, we can significantly enhance the predictive capability of the original CTR model. However, the added relevance network in the Retrieval Framework, due to its retrieval process (such as BM25 [30]), is difficult to deploy online due to its inference time, similar to other retrieval-based CTR models [25, 47]. To address this difficulty, we further build the Distill Framework to distill knowledge from the relevance network using shifting data, obtaining a parameterized student network called search-distill module. By aggregating the original CTR model and the search-distill module, we finetune the Distill Framework with unshifting data, e.g., the latest training data. This yields the final model that can be deployed online for light workload and efficient inference.

In sum, the technical contributions of this paper are threefold.

- We highlight the significant impact of the data shift phenomenon in recommendation systems, which is not addressable by simply increasing training data or the model capacity. To deal with this issue, we propose the Retrieval and Distill (RAD) paradigm, where the recommendation model is enhanced by incorporating shifting data into the model. To the best of our knowledge, RAD is the first paradigm that successfully utilizes shifting data for model training and lightweight deployment.
- As the rationale of RAD, we raise the theorem of Temporal Invariance of Association, which provides effective guidance on how to utilize shifting data.
- We design an innovative distillation approach that distills information from the retrieval framework of RAD. This approach optimizes both the time complexity and space complexity of the retrieval process to $O(1)$, which largely pushes forward the practical usage of retrieval-based models for online recommendation systems.

Extensive experiments on multiple real datasets demonstrate that RAD significantly improves the performance of the original model via effective training over shifting data. RAD offers a new paradigm in the field of recommendation systems, enabling models to train using both shifting and unshifting data, opening up a new pathway for enhancing the performance of CTR models by increasing the amount of training data.

2 PRELIMINARIES

In this section, we formulate the problem and introduce the notations. In the context of recommendation systems, we represent the dataset as a tabular structure, denoted by $\mathcal{D}$, comprising F feature columns $C = \{C_i\}_{i=1}^{F}$ alongside a singular label column $\mathcal{Y}$. The feature columns encapsulate attributes such as "Age," "City," and "Category." Each individual entry within table $\mathcal{D}$ corresponds to a sample s_z. A sample is characterized by an amalgamation of multiple features along with a label, articulated as $s_z = (x_z, y_z)$, where $x_z = \{c_i^z\}_{i=1}^{F}$. Accordingly, table $\mathcal{D}$ is defined as an ensemble of N samples, expressed as $\mathcal{D} = \{s_z\}_{z=1}^{N}$.

A click-through rate (CTR) model can be mathematically represented as a conditional probability, $P(y|x)$, where y is the event of a click, and x represents the features. The goal of the model is to accurately estimate this probability given the features x, which can include information about the item, the user, and the context.

Theorem 2.1. *Temporal Invariance of Association.*

In the paper, we model $P(y|x)$ as $P(y|x, x_s)$, where x_s represents documents similar to x within a fixed search space $\mathcal{E}$. Given that the probability distribution of $P(y|x, x_s)$ changes over time, while the probability distribution of $P(x_s|x)$ remains constant, we can utilize the data come from the time-varying $P(y|x, x_s)$ to train a model that represents $P(x_s|x)$. This process allows us to capture the temporal dynamics in $P(y|x)$ while leveraging the time-invariant properties of $P(x_s|x)$, thereby enhancing the model's adaptability to temporal changes in the future.

We partition the dataset by date into three segments. The last few days, denoted as $\mathcal{D}_{[t+1,t+\Delta]}$, serve as the test set, $\mathcal{D}_{\text{test}}$. Data within a window of d days prior to the test dates is considered unshifting data, which is $\mathcal{D}_{[t-d+1,t]}$ and is used as the training set, denoted by $\mathcal{D}_{\text{train}}$. The remaining portion, $\mathcal{D}_{(-\infty,t-d]}$, is designated as the search space, $\mathcal{D}_{\text{shifting}}$.

The **Retrieval and Distill (RAD)** framework comprises two components. The first component, the Retrieval Framework, employs a retrieval-based method to assist any original model in addressing the data shift issue. However, retrieval-based models face significant efficiency challenges. Therefore, the second component, the Distill Framework, utilizes distillation to eliminate the retrieval process. We will introduce these two components separately in the following.

Retrieval Framework. The objective of the Retrieval Framework is to employ a relevance network, which incorporates non-parametric retrieval, to assist any original model $F(x)$ in predicting the label. The formulation of the Retrieval Framework is as follows:

$$\hat{y_R} = f_\theta(F(x), x, R(x)), \tag{1}$$

where $R(x)$ is designed to fit $P(x_s|x)$ and is referred to as the *relevance network* and f_θ which approximate the $P(y|x, x_s)$ is data shift sensitive.

To obtain a model, $R(x)$, that approximates the probability distribution $P(x|x_s)$, we employ a Retrieval Framework, devoid of $F(x)$, named the teacher network. The formulation of the teacher network is as follows:

$$\hat{y_T} = T_\gamma(x, R(x)). \tag{2}$$

According to Theorem 2.1, The $R(x)$ component within T_γ, trained using shifting data, is invariant to data shift. Therefore, to obtain an $R(x)$ that has been trained with shifting data, T_γ can be minimized by the following loss function:

$$\mathcal{L}_{\text{pretrain}} = \sum_{x_z \in \mathcal{D}_{\text{shifting}}} \text{CE}(y^z, \hat{y}^z_T), \tag{3}$$

where CE is binary cross-entropy. This step obtains the model $R(x)$ that approximates $P(x_s|x)$.

Once we have obtained an $R(x)$ that has been pretrained with shifting data, we can utilize $R(x)$ in the training of the Retrieval Framework. We use unshifting data to minimize the following loss function:

$$\mathcal{L}_{\text{retrieval}} = \sum_{x_z \in \mathcal{D}_{\text{train}}} \text{CE}(y^z, \hat{y}^z_R), \tag{4}$$

where CE is binary cross-entropy. In this step, we obtain the Retrieval Framework, f_θ, which approximates $P(y|x, x_s)$.

Distill Framework. Although the Retrieval Framework can leverage shifting data to significantly enhance the original model, the deployment of retrieval-based methods presents substantial challenges in terms of storage space and inference time in an online environment. Therefore, we have designed the Distill Framework to eliminate the non-parametric search process.

Distill Framework can be formulated as:

$$\hat{y}_D = f'_\phi(F(x), x, r(x)), \tag{5}$$

where $r(x)$ is distilled from $R(x)$ pretrained using loss function (3), which we refer to as the *search-distill module*. Similar to f_θ, f'_ϕ approximates $P(y|x, x_s)$ and can also only be trained using unshifting data.

We assume

$$w_R = R(x), \tag{6}$$

where w_R is the last layer logits of $R(x)$ and assume

$$w_r = r(x), \tag{7}$$

where w_s is the last layer logits of $r(x)$. The loss function used to distill knowledge from $R(x)$ to $r(x)$ can be written as:

$$\mathcal{L}_{\text{KD}} = \sum_{x_z \in \mathcal{D}_{\text{shifting}}} \text{MSE}(r(x), R(x)) = \sum_{x_z \in \mathcal{D}_{\text{shifting}}} \text{MSE}(w^z_r, w^z_R), \tag{8}$$

where MSE is the Mean Squared Error loss function. Once we obtain the search-distill module, we can use unshifting data to derive a final Distill Framework using the following loss function:

$$\mathcal{L}_{\text{distill}} = \sum_{x_z \in \mathcal{D}_{\text{train}}} \text{CE}(y^z, \hat{y}^z_D), \tag{9}$$

where CE is binary cross-entropy loss.

3 THE RAD PARADIGM

In this section, the details of RAD will be presented. We start with an overview of the entire paradigm's structure at Section 3.1, followed by detailed expositions of the Retrieval Framework and Distill Framework in Sections 3.2 and 3.3, respectively.

3.1 Overview

Traditional CTR models have not accounted for the data shift issue, and they cannot enhance performance by merely adding shifting data [6, 13, 20]. Training only with unshifting data neglects a significant amount of information. Our proposed Retrieval and Distill paradigm aims to devise a universal solution that utilizes shifting data to enhance the performance of existing models. RAD is generally divided into two parts. The first part incorporates a Retrieval Framework to introduce shifting data into the datastore during training. Concurrently, as per theorem 2.1, we pretrain the model's relevance network with shifting data and then transfer it for use. This approach substantially improves the model's performance without altering its original structure. However, including a non-parametric search component in the relevance network poses deployment challenges for online recommendation systems. Therefore, the second part is a Distill Framework tasked with transforming the relevance network into a parametric neural network model. We distill a simple neural network (search-distill module) using shifting data and the relevance network as the teacher network. Ultimately, we merge the search-distill module with the original model and finetune it using unshifting data to yield the final online deployable model. The following sections will elucidate these modules in detail.

3.2 Retreival Framework

The Retrieval Framework aims to employ a relevance network that utilizes shifting data in conjunction with the original model for CTR prediction. According to theorem 2.1, the relevance network is temporal data shift-free; therefore, as an initial step, we pretrain the relevance network using shifting data with a teacher network, which is a Retrieval Framework that does not contain the original model. In the second step, we jointly train the relevance network and original model using unshifting data. Next, we will detail the neural network structure of the Retrieval Framework, followed by its training method.

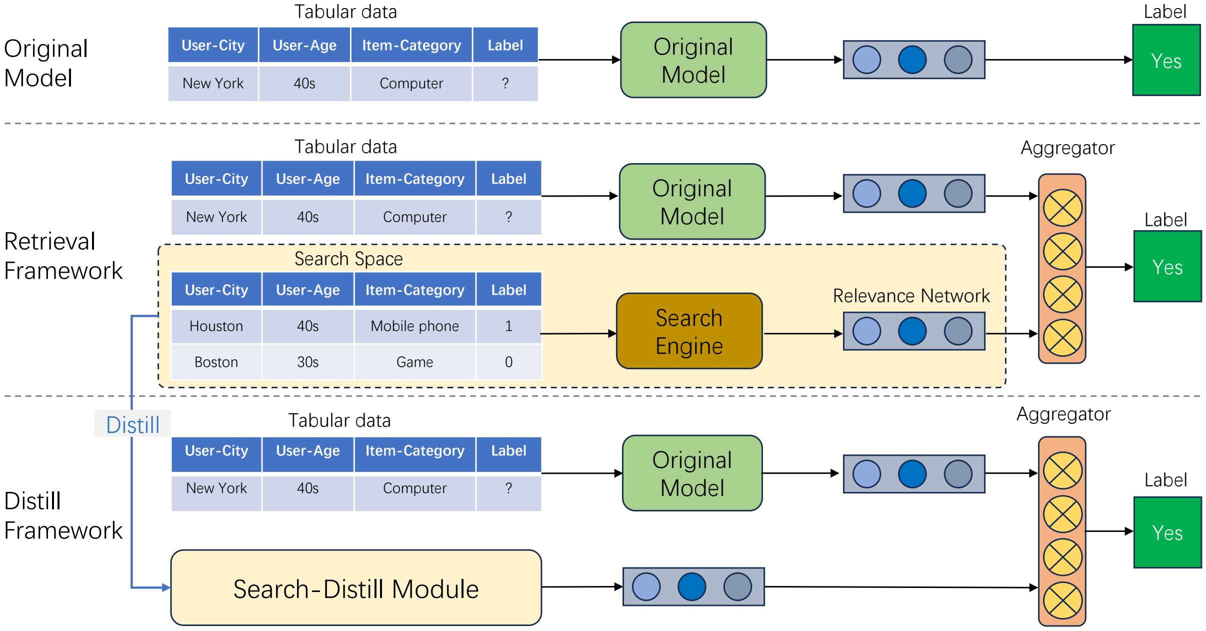

Figure 2: This figure illustrates the original model and its combination with the two components of the RAD paradigm. In CTR tasks, the original model typically takes tabular data as input and outputs a probability indicating the likelihood of a click. The RAD paradigm initially employs a retrieval framework to assist the original model in making predictions using shifting data. However, considering the performance requirements of the CTR model, we designed the RAD's distill framework to eliminate the search process in the retrieval framework, thereby meeting the online inference time requirements.

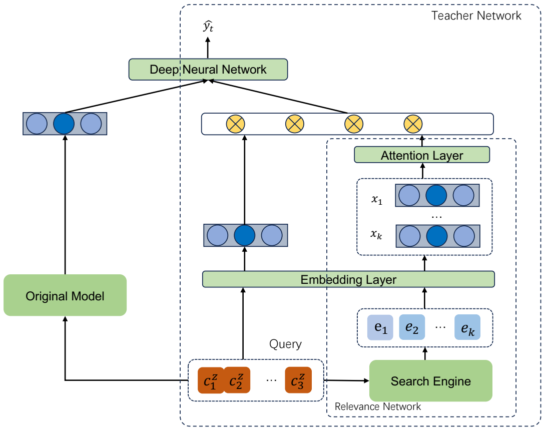

Figure 3: Retrieval framework.

3.2.1 *Framework Model Structure.*

The neural network structure of the Retrieval Framework can be seen in Figure 3. The Retrieval Framework aggregates the outputs of both the teacher network and the original model, where the teacher network specifically combines the original input with the relevance network. We shall now explicate the original model, relevance network, and teacher network.

Original Model. We denote s_z as any tabular data entry within the dataset $\mathcal{D}$, where the feature part of s_z is represented as x_z, and the label part as y_z, with $x_z = \{c_z^i\}_{i=1}^F$ comprising F features. Traditional CTR models take such an x_z as input and output a probability. We represent the original model as O, with the last layer logit denoted by w_o.

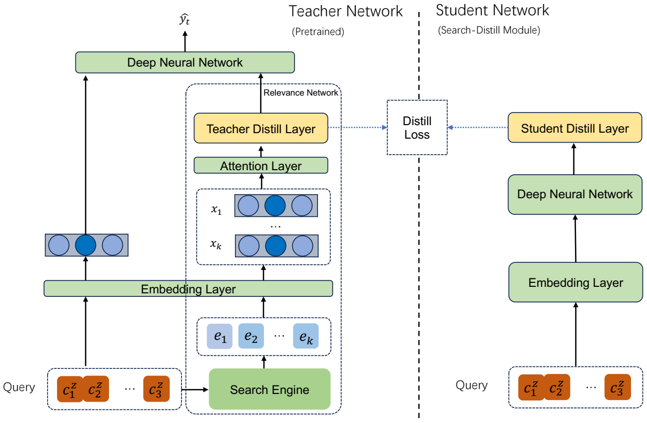

Figure 4: Distill Process.

In the traditional model, the probability prediction is given by

$$\hat{y}_O^z = O(x_z) = \text{sigmoid}(w_o), \tag{10}$$

where w_o is the input of the aggregation layer of the Retrieval Framework.

Relevance Network. The relevance network features an embedding layer shared with input x_z, a search engine, and an attention-based aggregation layer. Within our framework, we utilize shifting tabular data entries as the search space, with BM25 [30] serving as our search algorithm.

In the BM25 algorithm, we treat any sample x_z as the query and a key x_d in the datastore as the document. As such, the ranking

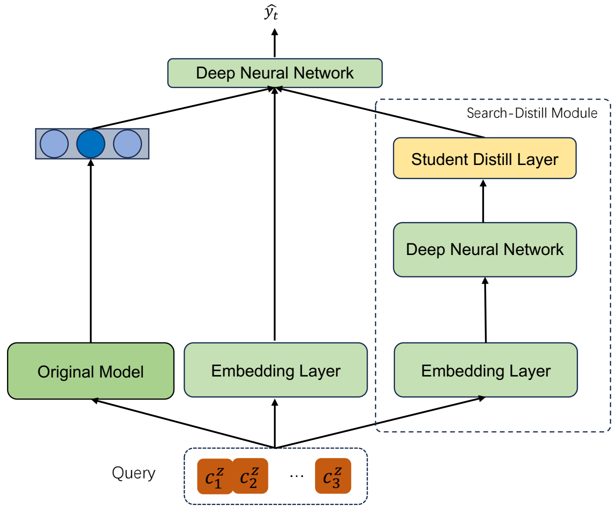

Figure 5: Distill Framework.

score can be calculated as

$$\text{RankScore}(x_z, x_d) = \sum_{j=1}^{M} \text{IDF}(a_j^z) \cdot \mathbf{1}(x_j^z = x_j^d), \tag{11}$$

$$\text{IDF}(x_j^z) = \log \frac{N - N(a_j^z) + 0.5}{N(a_j^z) + 0.5}, \tag{12}$$

where $\mathbf{1}(\cdot)$ is the indicator function, N is the number of data samples in $\mathcal{D}_{\text{shifting}}$, and $N(a_j^z)$ is the number of data samples that have the categorical feature value x_j^z. By ranking the candidate samples using BM25, we obtain the top K entries as the retrieved set $\mathcal{S}(x_z)$.

For each item within $\mathcal{S}(x_z)$, an attention mechanism is applied to calculate its aggregation parameter

$$\alpha_k = \frac{\exp(x_k^\top W x_z)}{\sum_{j=1}^{K} \exp(x_j^\top W x_z)}, \tag{13}$$

The elements of $\mathcal{S}(x_z)$ are then consolidated into a logit w_R :

$$w_R = \sum_{k=1}^{K} \alpha_k \cdot x_k, \quad s_k \in \mathcal{S}(x_z), \tag{14}$$

where w_s is the input of the aggregation layer of the teacher network.

Teacher Network. The teacher network is a Retrieval Framework that does not include the original model. From Figure 3, we can observe that the teacher network forms the right portion of the diagram. The teacher network serves as an intermediary bridge for training the relevance network. The teacher network takes x_z and the relevance network output as inputs, producing a probability as output. Drawing inspiration from RIM [25], we incorporate a factorization machine module within the aggregation layer of the teacher network. This aggregation layer is designed to effectively merge the input x_z with the aggregated w_R, thus improving the teacher network's capability to discover the hidden relationships between x_z and the returned $S(x_z)$ from the search engine.

3.2.2 Pretrain with Shifting Data. The relevance network models the probability distribution $P(w_R|x)$, where w_R is the logit representation of x_s. As established by theorem 2.1, the probability distribution represented by the relevance network is time-invariant, which allows us to pretrain it using shifting data. To obtain the relevance network, we initialize a separate teacher network. The teacher network is trained using the dataset $\mathcal{D}_{\text{shifting}}$ and with loss function (3). Although the separate teacher network trained in this manner performs poorly on the test set $\mathcal{D}_{\text{test}}$ due to data shift—often hovering around 0.5 in our experiments—the relevance network is transferable.

3.2.3 Finetune with Unshifting Data. After obtaining the pretrained relevance network, we integrate it with the original model and the target query x_z to form the Retrieval Framework. This framework is then trained using $\mathcal{D}_{\text{train}}$ with loss function (4), essentially finetuning it.

3.3 Distill Framework

Although the Retrieval Framework alone allows us to significantly enhance the performance of various existing CTR models by utilizing shifting data, this improvement depends on the relevance network's inclusion of BM25 search, which can be computationally intensive due to the typically large volume of $\mathcal{D}_{\text{shifting}}$ data and the often linear time complexity of the search process. This presents a substantial challenge for deploying the Retrieval Framework in an online environment. To optimize the time complexity of the Retrieval Framework, we propose a Distill Framework designed to distill the relevance network and remove its search component. The resulting module is termed the search-distill module.

Next, we start by delineating the structure of our model, followed by a description of the Distill Framework's training process. Within the Retrieval Framework's pretraining step, we achieve a pretrained relevance network. In the Distill Framework, this network undergoes distillation using $\mathcal{D}_{\text{shifting}}$. Subsequently, the distilled search-distill module is combined with the original model through an integrated multilayer neural network for aggregation and subsequent finetuning using $\mathcal{D}_{training}$.

3.3.1 Framework Model Structure. The neural network structure of the Retrieval Framework can be seen in Figure 5. The network architecture of the Distill Framework is similar to that of the Retrieval Framework. We replace the relevance network with the search-distill module.

From Figure 4, we can observe the form of the teacher network in the Distill Framework. The teacher network in the Distill Framework differs somewhat from that in the Retrieval Framework. We removed the cross layer from the teacher network in the Retrieval Framework to distill only the relevance network portion and added two neural network layers after the attention layer. We use this teacher network to obtain a relevance network for distillation.

The search-distill module is a multilayer neural network equipped with an embedding layer. Akin to the original model, the search-distill module takes the feature portion of tabular data x_z as its input. The output of the search-distill module is a logit of the same size as that of the final output from the relevance network.

3.3.2 Pretrain and Distill with Shifting Data. From theorem 2.1, we understand that the relevance network operates as a temporal data shift-free module. Like the relevance network, the search-distill module is also used to approximate the probability $P(x_s|x)$, which can be trained using shifting data. We first obtain the relevance network using the teacher network and loss function (3). Then, we employ $\mathcal{D}_{\text{shifting}}$ and loss function (8) to perform distillation on the final logit layer of the search-distill module. The distillation process can be observed in Figure 4. We employ the mean squared error loss function to align the final logit layer of the search-distill module with that of the relevance network.

3.3.3 Finetune with Unshifting Data. After obtaining a distilled search-distill module, we integrate the original model, the input x_z, and the search-distill module using a multilayer neural network for aggregation. This network is then trained with unshifting data and loss function (9), allowing us to finetune and enhance the original model with insights gleaned from shifting data without significantly increasing the deployment overhead of the original model.

4 EXPERIMENTS

In this section, we evaluate the RAD paradigm across multiple datasets using various traditional CTR models. We measure the efficacy of the RAD paradigm using the conventional CTR prediction task. We select two sets of classic CTR models: one representing factorization models and the other representing sequence prediction models. Each set of models is tested on three real-world datasets.

This section starts with five research questions (RQs) guiding the subsequent discourse.

- **RQ1** Does the RAD paradigm enhance the performance of existing CTR prediction models?
- **RQ2** Can distillation functions be applied to the distillation of non-parametric models?
- **RQ3** Can the relevance network, trained using shifting data, be transferred to predict future data?
- **RQ4** How does the RAD paradigm affect the run time of the original model?
- **RQ5** How do different distillation loss functions affect the RAD paradigm?

4.1 Datasets

In order to evaluate the performance of RAD, we conduct extensive experiments for CTR prediction tasks on three real-world large-scale recommendation datasets from Alibaba and Ant Group, i.e., Taobao, Tmall and Alipay[1]. These datasets consist of sequential user-item interactions in e-commerce scenes. Table 1 is the information for our split dataset.

4.2 Evaluation Metrics

In the CTR prediction task, the area under the ROC curve (AUC) and negative log-likelihood (LogLoss) are widely used metrics. AUC measures the probability that a randomly chosen positive sample will be ranked higher than a randomly chosen negative sample, while Logloss quantifies the difference between the predicted probability and the actual label. We choose these two metrics to evaluate the model performance.

Datasets	Fields	Split	# Instances	Start time	End time
Taobao	4	$\mathcal{D}_{\text{shifting}}$	145416702	2017-11-25	2017-12-01
		$\mathcal{D}_{\text{train}}$	2754415	2017-12-02	2017-12-02
		$\mathcal{D}_{\text{test}}$	2734177	2017-12-03	2017-12-03
Tmall	9	$\mathcal{D}_{\text{shifting}}$	68646472	2015-05-11	2015-10-31
		$\mathcal{D}_{\text{train}}$	6679610	2015-11-01	2015-11-10
		$\mathcal{D}_{\text{test}}$	7055119	2015-11-11	2015-11-12
Alipay	6	$\mathcal{D}_{\text{shifting}}$	82090360	2015-07-01	2015-10-31
		$\mathcal{D}_{\text{train}}$	264843	2015-11-21	2015-11-24
		$\mathcal{D}_{\text{test}}$	431646	2015-11-25	2015-11-30

Table 1: Datasets

4.3 Compared Methods

To evaluate the effectiveness of our proposed architecture, we choose six commonly used and strong models as backbones, which can be grouped into two classes.

4.3.1 Feature interaction based Models. The first class includes some traditional feature interaction based tabular models. DNN represents multilayer perceptron (MLP), a simple feedforward artificial neural network. DeepFM [7] is an architecture that combines the factorization machine and deep neural network. PNN [27] is a product-based neural network which captures both inter-field interactive patterns and high-order feature interactions.

4.3.2 Sequential based Models. The second class contains several sequential end-to-end deep models. Transformer [35] is a sequence-to-sequence model based on self-attention mechanism. Here, we use its encoder to model users' history behaviors. DIN [49] and DIEN [48]are two attention-based sequential models, while DIEN adds a recurrent neural network based on DIN.

4.4 Overall Performance (RQ1)

The results are listed in table 2. The results reveal the following insights. i) In our experiments, sequence models, except the original Transformer model, outperformed factorization models. This indicates that increasing data volume is beneficial for recommendation systems. ii) The Retrieval Framework utilizes shifting data as a search space and pretrains the relevance network with shifting data, significantly enhancing performance beyond the original model. This demonstrates the substantial benefit of shifting data for the original model. iii) The Distill Framework employs the distilled relevance network to assist the original model. While predictive performance surpasses the original model, it falls short of models augmented by the Retrieval Framework, suggesting that the distilled relevance network retains valuable information, albeit with less incremental information than the retrieval model. iv) The performance of feature interaction-based models within the Distill Framework has already surpassed that of the original sequence models. Moreover, as illustrated in Figure 7, the distillation method can enhance model performance without significantly increasing the inference burden

[1] https://tianchi.aliyun.com/dataset/x, where 'x' is '649', '42', and '53' for Taobao, Tmall, and Alipay, respectively.

Models		Taobao			Tmall			Alipay		
		LogLoss	AUC	Rel. Impr.	LogLoss	AUC	Rel. Impr.	LogLoss	AUC	Rel. Impr.
Original	DNN	0.6454	0.6744	-	0.4259	0.8890	-	0.6230	0.7011	-
	DeepFM	0.6591	0.6545	-	0.4264	0.8885	-	0.6375	0.6803	-
	PNN	0.6469	0.6787	-	0.4274	0.8949	-	0.6253	0.7130	-
	Transformer	0.6757	0.5990	-	0.4644	0.8609	-	0.6761	0.6523	-
	DIN	0.6267	0.7040	-	0.4301	0.8820	-	0.6257	0.7333	-
	DIEN	0.6328	0.6947	-	0.3930	0.9037	-	0.6295	0.7218	-
Retrieval	DNN	0.4525	0.8633	28.00%	0.2985	0.9446	6.26%	0.4897	0.8455	20.60%
	DeepFM	0.4566	0.8614	31.61%	0.3016	0.9443	6.27%	0.4694	0.8583	26.16%
	PNN	0.4576	0.8606	26.81%	0.3054	0.9468	5.80%	0.4429	0.8748	22.68%
	Transformer	0.4614	0.8557	42.86%	0.3459	0.9249	7.44%	0.5606	0.7984	21.23%
	DIN	0.5558	0.7906	12.30%	0.3239	0.9354	6.05%	0.5626	0.7826	6.72%
	DIEN	0.5733	0.7702	10.87%	0.3226	0.9371	3.10%	0.5810	0.7684	6.46%
Distill	DNN	0.7201	0.7224	7.11%	0.4100	0.8961	0.80%	0.5972	0.7423	5.88%
	DeepFM	0.6144	0.7192	9.88%	0.4108	0.8961	0.86%	0.5900	0.7532	10.72%
	PNN	0.6106	0.7269	7.10%	0.4061	0.9026	0.87%	0.5686	0.7713	8.17%
	Transformer	0.6263	0.6953	16.09%	0.4237	0.8940	3.85%	0.6227	0.7196	10.3%
	DIN	0.5956	0.7420	5.40%	0.3812	0.9152	3.76%	0.5927	0.7425	1.25%
	DIEN	0.6099	0.7176	3.29%	0.3690	0.9157	1.33%	0.6230	0.7562	4.77%

Table 2: Performance comparison of CTR prediction task. Rel.Impr means relative AUC improvement rate against each original model. Improvements are statistically significant with $p < 0.01$.

Models		Taobao		Tmall		Alipay	
		LogLoss	AUC	LogLoss	AUC	LogLoss	AUC
relevance network pretrained with $\mathcal{D}_{\text{train}}$	DNN	0.6210	0.7139	0.3373	0.9291	0.6227	0.7070
	DeepFM	0.6239	0.7109	0.3337	0.9320	0.6300	0.6966
	PNN	0.6188	0.7226	0.3272	0.9405	0.6063	0.7339
relevance network pretrained with $\mathcal{D}_{\text{shifting}}$	DNN	0.5224	0.8163	0.3141	0.9398	0.5006	0.8396
	DeepFM	0.5183	0.8169	0.3103	0.9405	0.4724	0.8564
	PNN	0.5087	0.8256	0.3179	0.9438	0.4446	0.8744

Table 3: Performance comparison of pretrained relevance network in Retrieval Framework.

Loss Functions		Taobao	Tmall	Alipay
Mean Squared Error loss	DNN	0.7224	0.8961	0.7423
	DeepFM	0.7192	0.8961	0.7532
	PNN	0.7269	0.9026	0.7713
Kullback-Leibler loss	DNN	0.7238	0.8937	0.7722
	DeepFM	0.7220	0.8940	0.7661
	PNN	0.7277	0.9013	0.7865
Cosine Similarity loss	DNN	0.7274	0.8936	0.7649
	DeepFM	0.7248	0.8940	0.7593
	PNN	0.7301	0.9030	0.7791

Table 4: Distill Framework AUC of different distillation loss functions.

4.5 Applying Knowledge Distillation to Non-Parametric Models (RQ2)

Knowledge distillation is traditionally employed for transferring knowledge between parametric models, typically from a complex, larger model to a simpler, smaller one. In this context, we extend the application of knowledge distillation to transfer knowledge from a non-parametric model (BM25 search) to a simple neural network. As illustrated in Table 2, distilling information from the retrieval-based relevance network into the search-distill module demonstrates that the distilled student model can still significantly enhance the performance of the original model. However, compared to the direct use of the non-parametric model, performance is considerably reduced. There is significant room for improvement in distilling information from non-parametric models into parametric models.

4.6 Temporal Invariance of Association (RQ3)

Table 3 showcases a group of experiments designed to validate the theorem 2.1 of Temporal Invariance of Association. The relevance network aids the original model in prediction in two ways: the additional information provided by the retrieved similar data x_s, and the information gain brought by pretraining with shifting data. To eliminate the influence of x_s on the results during testing, we added D_{train} pretraining for comparison. Similarly, to rule out the impact of finetuning on the relevance network, we fixed the parameters of the relevance network during the finetuning phase. By comparing the data in the table, we find that: i) The relevance network pretrained with shifting data performs better than the model pretrained only with train data. This indicates that the relevance network trained with shifting data indeed learns time-invariant information in the time-varying data distribution. Theorem 2.1 is validated. ii) The relevance network trained with $\mathcal{D}_{\text{shifting}}$ performs better because whether using $\mathcal{D}_{\text{shifting}}$ or $\mathcal{D}_{\text{train}}$, both are fitting $P(x_s|x)$, and shifting data has a much larger volume. Hence, the trained relevance network performs better. Thus, we obtain a network that enhances model training effectiveness by increasing the volume of data, unlike the network in figure 1, where increasing the volume of data would decrease model performance.

4.7 Performance Efficiency Study (RQ4)

Recently, a range of retrieval-based models have been introduced in the field of recommender systems, some of which utilize discrete features for retrieval, such as SIM [23] and RIM[25], while others employ continuous features, like DERT[47]. Due to the significant time overhead associated with retrieval, these models face challenges in being directly deployed online. In Figure 6, we compare three methods for augmenting relevant information. As RAD employs distillation to bypass the retrieval process, it incurs no retrieval time. On the other hand, directly aggregating the search-distill module with the original model increases the inference burden on the original model. However, as shown in Figure 7, the additional inference load introduced by incorporating the search-distill module is not significant, and the time overhead it introduces is considerably less than the burden added by using sequence models for model inference.

4.8 The Study of Distill Loss Functions (RQ5)

In Table 4, we compare the impact of various distillation functions on the Distill Framework. For simplicity, our experiments were conducted using only three models based on feature interaction. Our findings indicate that the performance enhancement provided by different distillation loss functions is broadly similar, suggesting that the quantity of information extracted primarily depends on the shifting data itself. However, slight variations exist in the performance of different loss functions across datasets. The Mean Squared Error Loss Function generally performs slightly better on Tmall, the Kullback-Leibler Divergence Loss Function shows slight superiority on Alipay, and the cosine similarity loss function performs marginally better on Taobao. When addressing different data distributions, selecting an appropriate distillation function tailored to the specific dataset can further enhance the distillation effect.

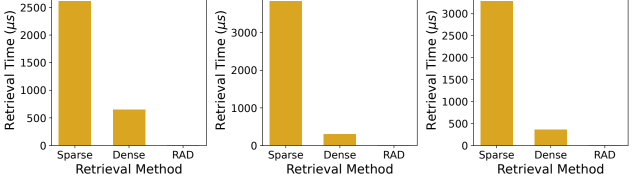

Figure 6: Comparison of Retrieval time.

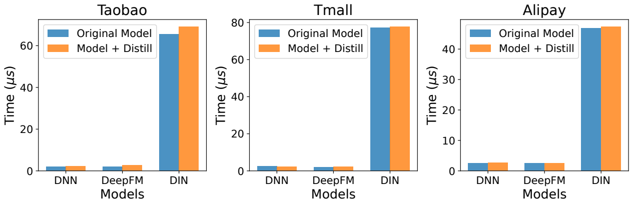

Figure 7: Comparison of Inference time.

5 RELATED WORK

Click-through rate (CTR) prediction is crucial in the fields of online advertising, recommendation systems, and information retrieval. Many deep learning models for CTR prediction primarily concentrate on investigating the feature interactions within a single instance [4, 7, 9, 18, 27]. The Wide&Deep network [2] constructs fully connected (FC) layers based on field feature embeddings to extract feature interactions implicitly. CCPM [19] employs convolution operators to probe local feature interaction patterns from a list of feature embeddings. PNN [27] suggests executing inner or outer product operations on each pair of field feature embeddings, followed by FC layers for further implicit feature interaction extraction. DeepFM [7] evolves from the factorization machine (FM) [29]. Recently, graph neural networks have also shown great potential in recommendation scenes by constructing data into graph structures and performing message passing [8, 17, 38–40]. Despite the numerous models proposed in this direction, the model capacity might be constrained as the model’s input is a single instance, necessitating the model to encode all knowledge into the parameters. Our work employs these deep models as the base model of the framework.

Retrieval-based methods for CTR prediction are typically derived from sequential recommendation tasks. where it’s common to use a recurrent neural network (RNN) to construct user profile representations for recommendation predictions. However, when the user’s historical behavior sequence becomes significantly long, RNN architectures struggle to memorize and utilize early behavior patterns [23]. Therefore, retrieval methods serve as a practical alternative to lengthy recurrence and attention operations. UBR [26] creates a search index from the user’s historical sequence data and employs reinforcement learning to generate a search query based on current prediction data. The retrieved data is then aggregated through an attention operation and input into an MLP for label prediction. SIM [24] executes both hard search based on item category information and soft search based on item embedding vectors, using the DIEN model [48] as the backbone. RIM [25] expands the research scenario from sequential recommendation to generic tabular data prediction. The query is the target row data, and the retrieved data instances are the relevant rows in the table, with BM25 [30] serving as the predefined relevance function between two rows. DERT [47] is the pioneering work to incorporate dense retrieval into recommendation systems. Retrieval-enhanced machine learning is now considered a significant frontier in machine learning research [45].

Knowledge Distillation (KD) [12] seeks to transfer the ’dark knowledge’ from a teacher model to a student model and is extensively used in model compression and knowledge transfer [5]. Recent studies have also employed KD in various recommendation tasks, such as click-through rate (CTR) prediction [51], topology distillation [14], and multi-task transfer [41]. Prior theoretical analyses [1, 22, 33] have indicated that the teacher's predictions, also referred to as soft labels, are typically viewed as more informative training signals compared to binary hard labels. KD can adaptively adjust the training dynamics of the student model based on the values of the soft labels. Knowledge distillation is typically employed to transfer knowledge from a larger parameterized model to a smaller one. Recent studies have applied knowledge distillation to transfer knowledge from KNN to parameterized models [42, 46], demonstrating the potential of knowledge distillation in non-parametric models.

6 CONCLUSION

In current recommendation systems, temporal data shift poses a significant challenge. The presence of data shift prevents the system from simply enhancing the CTR model's adaptability to new data by adding more training data. We observed that although the correlation between features and labels in recommendation systems changes over time, if a fixed search space is established, the relationship between the data and the search space remains invariant. Therefore, we designed a framework that uses retrieval techniques to leverage shifting data for training a relevance network. However, due to the use of BM25 as a retrieval method, this framework is challenging to deploy in online recommendation systems. We then designed a distillation method using knowledge distillation to transfer knowledge from the relevance network to a parameterized module, the search-distill module. We refer to this entire process as the Retrieval and Distill paradigm (RAD). With the RAD paradigm, we have an effective method for leveraging shifting data to enhance the performance of CTR models. In future research directions, we aim to incorporate a wider variety of data into the CTR model using RAD. On the other hand, enhancing the performance of the distillation method is also a significant area of focus.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Yunqing Zhao, and Ngai-Man Cheung. 2022. Revisiting Label Smoothing and Knowledge Distillation Compatibility: What was Missing?. In *International Conference on Machine Learning*. PMLR, 2890–2916.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *1st DLRS workshop*. 7–10.

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[4] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems* 34 (2021), 18932–18943.

[5] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[6] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2493–2500.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. *IJCAI* (2017).

[8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

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

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

[11] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 241–248.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[13] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM conference on recommender systems*. 43–50.

[14] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2021. Topology distillation for recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 829–839.

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

[16] Petros Katsileros, Nikiforos Mandilaras, Dimitrios Mallis, Vassilis Pitsikalis, Stavros Theodorakis, and Gil Chamiel. 2022. An Incremental Learning framework for Large-scale CTR Prediction. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 490–493.

[17] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *CIKM*.

[18] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic Feature Interaction Selection in Factorization Models for Click-Through Rate Prediction. *KDD* (2020).

[19] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A convolutional click prediction model. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 1743–1746.

[20] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G Azzolini, et al. 2019. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091* (2019).

[21] Danni Peng, Sinno Jialin Pan, Jie Zhang, and Anxiang Zeng. 2021. Learning an adaptive meta model-generator for incrementally updating recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 411–421.

[22] Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. In *International conference on machine learning*. PMLR, 5142–5151.

[23] 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.

[24] Pi Qi, Xiaoqiang Zhu, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[25] 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.

[26] Jiarui Qin, W. Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Y. 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*.

[27] 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. *TOIS* 37, 1 (2018), 1–35.

[28] 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. *SIGIR*.

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

[30] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at TREC-3. *Nist Special Publication Sp* 109 (1995), 109.

[31] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 255–262.

[32] 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.

[33] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. 2020. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532* (2020).

[34] 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.

[35] 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).

[36] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *ADKDD*. 1–7.

[37] 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.

[38] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[39] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[40] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[41] Chenxiao Yang, Junwei Pan, Xiaofeng Gao, Tingyu Jiang, Dapeng Liu, and Guihai Chen. 2022. Cross-task knowledge distillation in multi-task recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4318–4326.

[42] Zhixian Yang, Renliang Sun, and Xiaojun Wan. 2022. Nearest neighbor knowledge distillation for neural machine translation. *arXiv preprint arXiv:2205.00479* (2022).

[43] Mao Ye, Ruichen Jiang, Haoxiang Wang, Dhruv Choudhary, Xiaocong Du, Bhargav Bhushanam, Aryan Mokhtari, Arun Kejariwal, and Qiang Liu. 2022. Future gradient descent for adapting the temporal shifting data distribution in online recommendation systems. In *Uncertainty in Artificial Intelligence*. PMLR, 2256–2266.

[44] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 582–590.

[45] Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. 2022. Retrieval-Enhanced Machine Learning. *arXiv preprint arXiv:2205.01230* (2022).

[46] Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Zhen Li. 2023. Decoupled Non-Parametric Knowledge Distillation for end-to-End Speech Translation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[47] 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.

[48] 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.

[49] 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 *KDD*.

[50] Jieming Zhu, Guohao Cai, Junjie Huang, Zhenhua Dong, Ruiming Tang, and Weinan Zhang. 2023. ReLoop2: Building Self-Adaptive Recommendation Models via Responsive Error Compensation Loop. *arXiv preprint arXiv:2306.08808* (2023).

[51] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR prediction via knowledge distillation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2941–2958.