Retrieval and Ranking Alignment for Unfamiliar Recommendation in Long-Video Streaming Platform

Qidi Xu
qidi.xu@disneystreaming.com
Disney Streaming
Beijing, China

Pengyu Zhao
pengyu.zhao@disneystreaming.com
Disney Streaming
Beijing, China

Chunxu Xu
chunxu.xu@disneystreaming.com
Disney Streaming
Beijing, China

Liang Chen
liang.chen@disneystreaming.com
Disney Streaming
Beijing, China

ABSTRACT

The unfamiliar recommendation in the long-video streaming platforms is responsible for finding relevant videos that the users have never watched before to explore the users' potential interests, which usually follows the multi-stage paradigm, including a retrieval (candidate generation) stage that recalls hundreds of items from the entire corpus via a single objective and a ranking stage that sorts the retrieved outcomes based on multiple business objectives such as click-through rate (CTR), viewing time, and diversity. The existing retrieval methods are mainly sub-optimal due to the discrepancy between retrieval and ranking in the model capacity, training distribution, and objective. Meanwhile, the commonly adopted CTR or session-level watch minutes are inadequate to characterize the long-term patterns in the unfamiliar recommendation. In this paper, we propose a simple yet effective retrieval and ranking alignment (a.k.a, **RRA**) to solve the problems. Specifically, Discovery-induced Watching Minutes (DWM) is introduced as an additional objective to model the user's long-term engagement in the entire platform, which attributes the future viewing time within one week starting from the initial playback in the unfamiliar collections. To fill the discrepancy between retrieval and ranking stages, RRA coordinates the retrieval and ranking model with the partially-shared architecture that employs the shared embeddings and bottom layers to allow for implicit knowledge transfer, and then introduces various losses for aligning retrieval and ranking through synchronous training. Specifically, RRA proposes a multi-class classification loss to minimize the cross-entropy between training and ideal distribution and implicitly imitate the ranking outcomes with the pseudo labeling; a masked binary loss to align the retrieval distribution and objective with ranking meanwhile coordinates the logits range; a distillation loss to explicitly fill the gap of model capacity as well as the abovementioned components through knowledge transfer. The offline and online experiments demonstrate that RRA achieves superior performance compared to the other methods without introducing any serving overhead, indicating the effectiveness and efficiency of the design. RRA has been deployed into a real-world large-scale long-video streaming platform and brought significant viewing time lift for tens of millions of users every day.

KDD '23, Aug 6 – Aug 10, 2023, Long Beach, CA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Retrieval, Multi-Task Learning, Long-Video Recommendation, Knowledge Distillation

ACM Reference Format:
Qidi Xu, Pengyu Zhao, Chunxu Xu, and Liang Chen. 2023. Retrieval and Ranking Alignment for Unfamiliar Recommendation in Long-Video Streaming Platform. In *Proceedings of Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Long-video streaming platforms such as Netflix, Hulu, Disney+, and iQiyi are widespread in the Internet era, serving millions of users worldwide every day. Different from the short-video platforms like TikTok and Kwai that present the user-generated content to the users with the vertical information flow, the long-video streaming platforms provide the "long" videos (typically with a viewing time of more than 30 minutes) of professional-generated high-quality **shows**, including TV series, movies, and live events, to the paid subscribers. The contents in the long-video streaming platforms are usually delivered by the row-wise **collections** with different themes, which can be roughly categorized into the **familiar** collections with contents that the user has watched before and **unfamiliar** collections with contents the user has never watched. Once the user has consumed the content in the unfamiliar collections, the item will disappear from the unfamiliar scenarios and only be forwarded by the familiar collections. Thus, familiar collections like "Continuous Watching" are responsible for capturing the repeated, continuous, and subsequent user behaviors (e.g., a user may watch a movie more than once, finish the uncompleted watches, or consume the subsequent episodes of the same TV series); in contrast, the unfamiliar collections such as "For You" are

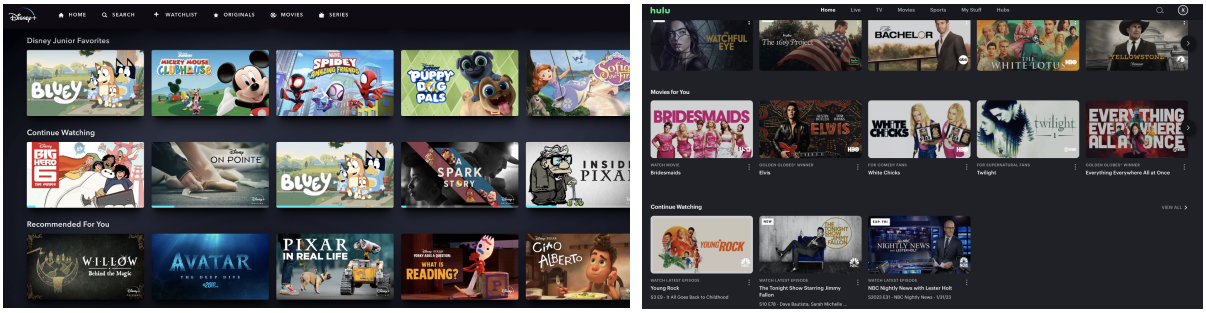

Figure 1: The homepages of the long-video streaming platforms Disney+ and Hulu, which contain both familiar collections like "Continuous Watching" and unfamiliar collections like "Recommended For You". The recommendation algorithms will act on the ordering of each show within the collections as well as the ordering of different collections.

responsible for discovering user's potential interests, which is crucial because they influence the user consumptions across various scenarios in the platform.

The collections in the long-video streaming platforms are powered by the recommender system for presenting personalized experiences, which could improve the efficiency of content discovery, reduce user churn, and increase the platform profit. The industrial recommendation usually follows the multi-stage paradigm, where a retrieval (candidate generation) stage is placed at the bottom of the system to roughly filter out hundreds or thousands of items from the massive content, afterwards, the cascaded ranking stages will give the precise scoring on each retrieved outcome. In this paper, we focus on the **retrieval** of the **unfamiliar** recommendation in the long-video streaming platform, as the bottom-layer candidate generation determines the lower bound of the overall system. The industry [4, 7, 9, 14] has proposed a lot of practical approaches for item retrieval and achieved remarkable success on various applications. However, those methods may lead to suboptimal performance in the considered task due to the two reasons described below.

Discrepancy between retrieval and ranking. The industrial recommendation follows the multi-stage paradigm. Nevertheless, the candidate set generated by the retrieval model is usually suboptimal and inconsistent with the final orders sorted by the ranking model due to the discrepancy among the following aspects:

- **Training distribution**: The retrieval model regularly uses the samples with the click (or playback) for model training, and regards the clicked item as positive and other items in the entire corpus as negative to mitigate the distribution gap between training and inference. On the other hand, the ranking model regularly adopts the impression samples for model training with the clicked items as positive and impressed not clicked items as negative (hard negative) to identify the subtle differences among the items with high interactive probability, which are closer to the distribution of online ranking that only performs the predictions on the retrieval outcomes. Though the retrieval-ranking pipeline can compensate for each other, it is usually hard for the retrieval model to distinguish the relative order among the items with strong user intents and thus it can not feed the ideal outcomes to the subsequent stages. Some recent works [7, 9] propose the hard negative minings in the retrieval stage to help better differentiate the clicked items from other top-ranked items, whereas we empirically found they failed to improve the performance in the long-video recommendation scenarios.
- **Objective**: The industrial retrieval [1, 4, 7, 9] mostly employ a single objective of click-through rate (CTR) to efficiently serve the model online based on the built index [9–11]. In contrast, the ranking model [17, 24, 35] mainly considers multiple objectives such as CTR, watch minutes, and diversity, that would finally be combined via the aggregation function at the inference time. The objective discrepancy can cause huge disparity, e.g., CTR prediction from the retrieval stage commonly deviates from the ones from the watch minute prediction, not along the hybrid scores of multiple objectives.
- **Model capacity**: The retrieval stage generally employs a relatively simple model (e.g., two-tower model) to efficiently select the candidates from the entire corpus, while the ranking model utilizes a more complex model with a bundle of interactive features for modeling the precise user-item preference. The difference in the model architecture leads to a considerable gap between the two stages.

Inadequate objective. The previous works [4, 9] tend to employ CTR or session-level watch minutes for optimizing the retrieval model. However, those metrics only characterize the short-term patterns within the unfamiliar collections, while neglecting the repeated, continuous, and subsequent behaviors in the familiar collections of the shows discovered and elicited by the unfamiliar collections, which usually account for the majority of user consumption. Therefore, the behaviors in the familiar collections should also be considered for modeling the entire user lifecycle and facilitating platform-level optimization.

In this paper, we introduce the **Retrieval and Ranking Alignment** (RRA) to explore the user's long-term preference and mitigate the discrepancy between retrieval and ranking stages for unfamiliar collections in the long-video streaming platforms. Specifically, RRA first incorporates the short-term objective CTR with a newly proposed Discovery-induced Watching Minutes (DWM) for modeling the long-term influence of the unfamiliar recommendation, which

attributes the future viewing time within one week starting from the initial playback in the unfamiliar collections over the entire platform. To achieve the alignment between retrieval and ranking, RRA proposes a hybrid loss for retrieval optimization based on the partially-shared architecture that coordinates the retrieval and ranking models, where the embedding layer and bottom behavior modeling layer are shared by two models whereas the top multi-task predictions are separated to produce the independent DWM and CTR scores. Given the predicted DWM and CTR, RRA proposes various losses to mitigate the discrepancy through synchronous training. Specifically, a weighted multi-class cross-entropy is applied on the impressions loggings as the basic classification loss to discriminate the easy negative, hard negative, and clicked items, which can be seen as the combination of the vanilla item retrieval task and the pseudo labeling from the entire recommender system. Then, a masked binary loss is introduced to explicitly fill the distribution and objective gaps meanwhile coordinating the logits scale by performing the identical ranking loss on the masked channels of retrieval outputs. Finally, RRA utilizes the soft cross-entropy between ranking and retrieval outputs on the target shows to directly transfer the knowledge from ranking to retrieval, which implicitly aligns the model capacity, training distribution, and objectives of the two stages.

The extensive offline experiments over the real-world long-video recommendation datasets show that RRA largely outperforms the baseline methods over the accuracy and retrieval-ranking consistency metrics without introducing any serving overhead, demonstrating the effectiveness and efficiency of the design. Moreover, online A/B experiments exhibit that RRA can even bring a significant lift of platform-level viewing time, indicating the necessity of adopting DWM optimization and aligning retrieval and ranking models in the unfamiliar recommendation of the long-video streaming platform. RRA has been deployed into the "TV For You" and the "Movies For You" collections of Hulu, empowering these unfamiliar collections for tens of millions of subscribers every day.

2 RELATED WORKS

Industrial Recommendation Systems. Promoted by a fast-growing deep learning methodology, deep recommendation models have been extensively applied in real-world recommendation practice including long-video recommendations. YouTube DNN [4] and Wide&Deep [3] are two milestones of the large-scale industrial deep recommendation system, which capture underlying relationships between users and items with Embedding&MLP paradigm. Particularly, [4] introduces the cascaded paradigm in the deep learning recommender system, which decomposes the pipeline into the retrieval and ranking stages. The successive works mainly consider the individual optimization of retrieval [1, 2, 14, 16, 18, 19, 28], ranking models [17, 24, 36, 37] or pre-ranking (modules in between) [27]. Different from these works, we propose RRA to solve the discrepancy between retrieval and ranking stages through alignment strategies and introduce the long-term objectives as the surrogate of the platform metrics for unfamiliar recommendations in the long-video streaming platform.

Fill the Gap among Different Stages in Recommendation. The industrial recommendation mainly employs the multi-stage paradigm to gradually narrow the candidate set provided to the users. As discussed in Sec.1, different stages in the recommender system naturally contain discrepancies of varying degrees, therefore, the existing literature proposes various techniques to fill the gaps. Knowledge distillation [25, 29, 34, 38] is commonly adopted for aligning the ranking model and pre-ranking model on the point-wise ranking predictions. [21] trains cascaded ranking models with a joint framework to imitate the online serving, which sequentially generates the candidate sets for all stages and then aligns the predictions of the bottom stages (retrieval) to the upstream stages (ranking). However, these works are difficult to be applied in the industrial system given the missing target of the retrieval input or the large computation overhead. Different from these methods, we propose RRA with considerable computation overhead that employs the ranking outputs to guide retrieval optimization through synchronous training.

3 PRELIMINARY

The recommendation in the unfamiliar scenarios (unfamiliar recommendation) of the long-video streaming platform is responsible for presenting the **shows** (series, movies, live events) that the users have never watched before to discover the potential user interests, which usually includes a retrieval stage and a ranking stage. The retrieval stage generates K shows with the highest preferences for each requested user as the retrieved result and passing it to the ranking model, which is usually formulated as a user-item similarity problem, i.e., for an input user u, calculate her or his preference for each candidate in the item set I. The item set I can be either the entire corpus or a subset of them, depending on the upstream business logic. To calculate the similarity between users and items, the **two-tower** architecture is usually applied in the industry [4, 7] that computes the separate embedding vectors through the user and item towers, and adopts the cosine similarity or inner product on the two-tower outputs to generate the final user-item preferences. Different from retrieval, ranking utilizes a more complex model to estimate the user's multiple intentions on the retrieval outcomes. For each user u, the ranking model predicts t scores $s_1^i, s_2^i, ..., s_t^i$ for each candidate item i, and then sorts the candidates based on a mixed formulation.

4 RETRIEVAL AND RANKING ALIGNMENT

In this section, we will first introduce Discovery-induced Watching Minutes (DWM) as the long-term objective for the unfamiliar recommendation in the long-video streaming platforms. Then, we will show the partially-shared architecture that coordinates the retrieval and ranking models in RRA, as well as the various losses on CTR and DWM predictions during synchronous optimization to achieve retrieval and ranking alignment. We will also describe the online serving of RRA in the production environment. Finally, we will consider two alternative distillation-based methods that mitigate the gap between retrieval and ranking stages, and show that they are either comparable or inferior to the proposed method.

4.1 Discovery-induced Watching Minutes

The existing methods [4, 9] mainly employ CTR or session-level watch minutes as the retrieval objectives. However, our previous

experiments suggest that optimizing short-term objectives in the considered scenario can bring more title consumptions and watch times within the unfamiliar collections, but could not bring much lift in the platform-level metrics. Based on the data analysis, we empirically found the inconsistency between the unfamiliar and platform metrics is due to the fact that the employed optimization objectives neglect the repeated, continuous, and subsequent watches in the familiar scenarios of the shows discovered and elicited by the unfamiliar collections which usually occupy much more time than the consumptions in the unfamiliar collections (remember when the shows are previously consumed by the user, they would no longer appear in the unfamiliar collections). For example, users usually watch a portion of the recommended videos (especially movies) within the unfamiliar collection, and then continue to consume the rest of the content in the familiar scenarios as the content lengths in the long-video streaming platforms are much longer than the other applications; they mostly watch the subsequent episodes in the "Continuous Watching" on the series previously found by the unfamiliar collections; they sometimes re-watch the event or movies repeatedly in the familiar collections with the first playback derived by the unfamiliar collections.

Facing the facts, we propose the **Discovery-induced Watching Minutes** (DWM) as the long-term objective in the unfamiliar recommendation, which attributes the future viewing time over the entire platform within one week starting from the initial playback on the shows discovered by the unfamiliar collections. We use one week as the attribution window because we found that the viewing time within seven days is highly correlated with the overall consumption across longer periods, e.g., one month. DWM considers the long-range platform-level user behaviors and thus can be regarded as the surrogate of the platform metrics. We append DWM as the complement to the short-term objectives (CTR in this paper, other metrics can also be easily adapted to the proposed framework) for both retrieval and ranking.

4.2 Model Architecture

4.2.1 Input Features. RRA utilizes various common features as the model input for retrieval and ranking stages, as shown in Figure 2.

- **User behavior features**. Behavior features are the most important ones for user portrait in the long-video recommendation. To isolate the semantics from different show types (series and movies), we extract the separated watch history on TV and movies, and append the side information on each behavior, e.g., episode watch count and viewing time. In practice, we set the behavior sequence length l to 100. More than watch history, we also use like, dislike, and saved items to capture users' explicit preferences. Those features are also divided by the show type.
- **User demographic features**. User demographic features depict the user attributes as well as the coarse-grained interest extracted by feature engineering. RRA employs the age, gender, and user preferences on the genres and tags to characterize the user's general interests, which are useful for the user cold start.
- **Item features**. RRA employs the sparse show ID as the main feature. Though other side information can also be used in RRA, e.g., genres, tastes, and titles, we only consider show ID for simplicity.

More than the common features, the ranking model also possesses interactive features like "user watch count on the target show" that can not be directly used by the retrieval model. Nevertheless, the retrieval model can still benefit from the knowledge implicitly distilled through the RRA synchronous training described later.

4.2.2 Embedding Layer. RRA indexes the input features into high-dimensional sparse vectors via one-hot or multi-hot encoding. Each feature is associated with an embedding table, and the sparse IDs of each sample are embedded by the table lookup. The embeddings are shared by ranking and retrieval models to avoid excessive parameters and repeated calculations meanwhile enabling the implicit knowledge transfer between the two stages.

4.2.3 Bottom Layer Behavior Modeling. The embedded features are fed to the bottom behavior modeling layers shared by retrieval and ranking stages. The watch behavior show ID embeddings are integrated with the corresponding side information embeddings through multiplication as the input of the self-attention modules, which follows the standard bidirectional Transformer [26] design that comprised of the stack of multi-head self-attention layer and the feed-forward networks. The outputs of the self-attention modules are aggregated by the sum pooling. For the other behavioral features of like, dislike, and saved, we utilize the sum pooling on the ID embeddings as they only provide limited information gain due to the sparse occurrence in the long-video streaming platform. Note that the retrieval model can benefit from the training course of the ranking model through the gradient updates on the shared parameters. We will empirically justify the phenomenon in Sec.5.

4.2.4 Top Layer Multi-Task Prediction. The retrieval and ranking model follow a similar architecture in the top layer to generate the separate multi-task representations for both CTR and DWM objectives, as illustrated in Fig.2. Specifically, RRA performs the explicit feature interactions via the product-based neural networks (PNN) [22] and then concatenates the outputs to the raw features as the input of the upper layers. To model CTR and DWM simultaneously, RRA employs the Multi-gate Mixture-of-expert (MMoE) [17], which transforms the input features to various subspaces to learn discriminative semantics and then mixes the experts' outputs through the individual softmax gates to produce the task-aware representations, that would finally be processed by corresponding non-linear layers to generate the CTR-oriented and DWM-oriented representations. As the retrieval model adopts the two-tower architecture, the multi-task layer receives the user-side inputs, i.e., behavioral representations and user features, to generate the CTR and DWM user embeddings, denoted by u_{ctr} and u_{dwm}. The user embeddings and item embedding are multiplied to compute the CTR and DWM prediction vectors $\hat{s}_{ctr}$ and $\hat{s}_{dwm}$. For the ranking model, user features and behavioral features are fused along with the item features and interactive features as the input of the PNN and MMoE model to directly calculate the CTR and DWM prediction scores on the target item i, i.e., s^i_{ctr} and s^i_{dwm}. The multi-task prediction of the retrieval model fundamentally mitigates the objective divergence from the ranking model.

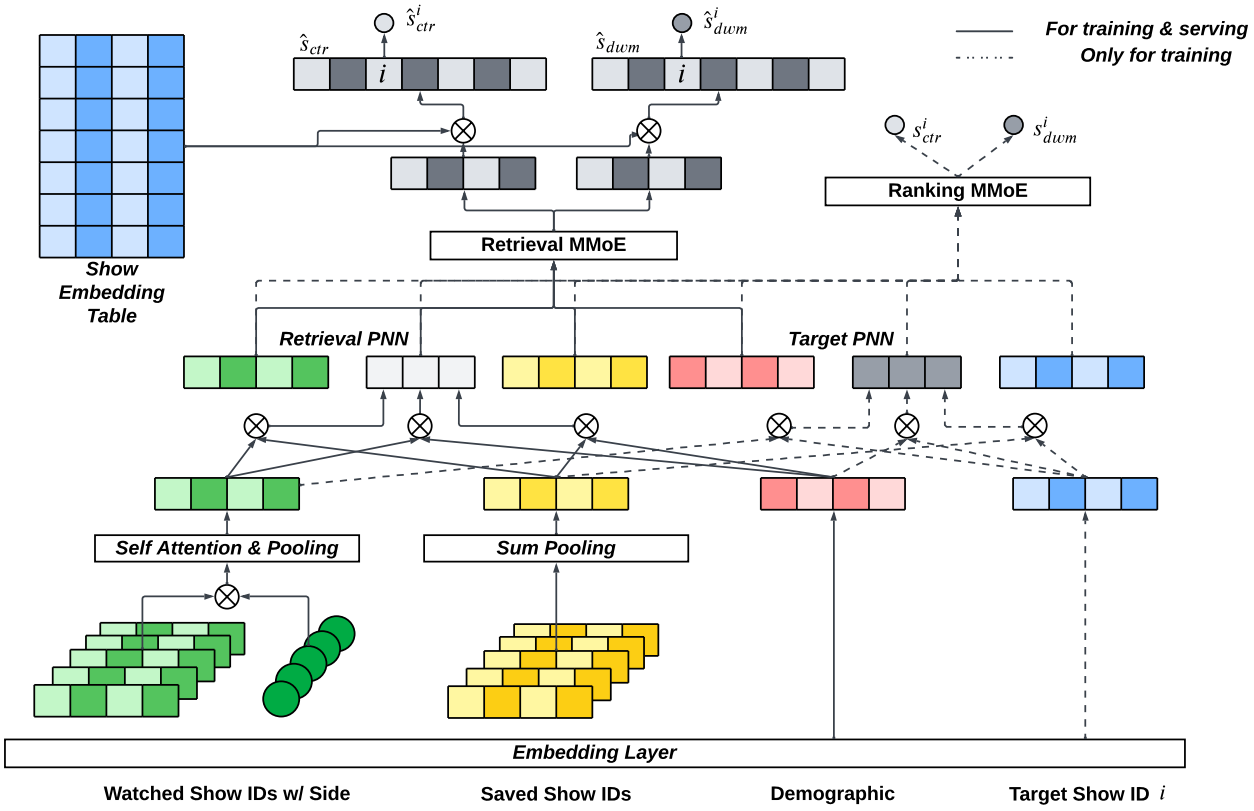

Figure 2: The partially-shared model architecture of RRA. RRA employs the shared embedding layer, shared bottom-layer behavior modeling, and individual top-layer multi-task predictions to generate the CTR and DWM scores for retrieval and ranking models. During online serving, RRA only computes the retrieval part and thus it does not introduce any additional computation overhead compared to the ordinary retrieval model.

4.3 Loss Constructors

To align retrieval and ranking, RRA synchronously trains both models and proposes a hybrid objective by summing up various paralleled CTR and DWM losses. The losses can be divided into four groups, and each group contains two loss functions corresponding to the CTR and DWM objectives.

4.3.1 Binary loss for ranking. For the ranking training, RRA employs the **impression** samples following the common practice, where **impressions with clicks** are considered as positive, and the other **impressions** are regarded as negative. The CTR prediction employs the simple binary log loss for optimization:

$$\mathcal{L}^{ctr}_{rank} = -y^i_{click} \log(\sigma(s^i_{ctr})) - (1 - y^i_{click}) \log(1 - \sigma(s^i_{ctr})), \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid activation function, i is the index of target item, y^i_{click} is the click label of the sample, and $\sigma(s^i_{ctr})$ is the corresponding prediction score. For the DWM prediction task, we apply the same log loss in Eqn.1 with the weight calculated from the DWM value w_{dwm}:

$$\mathcal{L}^{dwm}_{rank} = -w_{dwm} y^i_{click} \log(\sigma(s^i_{dwm})) - (1 - y^i_{click}) \log(1 - \sigma(s^i_{dwm})). \quad (2)$$

The logits optimized by Eqn.2 are proportional to the expected watch time of each sample [4].

4.3.2 Multi-class classification loss for retrieval. Different from the existing industrial retrieval [4, 32] that employs the **click** samples, RRA trains the model on the **impression** samples with the **impressed** shows as positive and the other shows in the candidate set as negative. The CTR task employs a weighted multi-class softmax cross-entropy loss:

$$\mathcal{L}^{ctr}_{cls} = -w_{ctr} \log \text{softmax}(\hat{s}_{ctr})^i, \quad (3)$$

where $\text{softmax}(\hat{s}_{ctr})^i$ is the retrieval's prediction on the impressed show i normalized by the softmax function. Other than equally handling all impressions, we differentiate the retrieval samples by assigning different weights: If the user clicks and watches the show, w_{ctr} is set to 1; if the user only impresses the show, the w_{ctr} is set to α $(0 < \alpha < 1)$. Note that if $\alpha = 0$, the objective would be identical to the vanilla retrieval task [4] with only the click samples as positive; if $\alpha = 1$, the objective would be the same as distilling the knowledge from the entire recommender system (especially the ranking model) via the pseudo labeling strategy [9, 13], which learns to return the same set of results that ranked higher by the ranking model and thus

improves the consistency between the retrieval and ranking stage. Hence, the retrieval model trained by Eqn.3 could discriminate the random (easy) negative from hard negative samples meanwhile maintaining the capability to distinguish the hard negative samples from the clicked items. More than the CTR task, the retrieval model also takes the DWM as the reward on the **clicked** items to optimize the expected viewing time with the REINFORCE algorithm:

$$\mathcal{L}_{cls}^{dwm} = -w_{dwm} \log \text{softmax}(\hat{s}_{dwm})^i \tag{4}$$

Eqn.4 could rank the shows that attribute to the longer discovery playback higher, which matches the ranking objective of Eqn.2. It is worth noting that we do not perform negative sampling on the candidate set during training as the complete similarity calculation is affordable owing to the limited cardinality (roughly tens of thousands) of the recommended corpus in the long-video streaming platform. We also compare the proposed all-negative strategy with the other sampling-based strategies in Sec.5.3.3, where we found that the all negative brings better retrieval performance.

4.3.3 Masked binary loss for retrieval. RRA also employs the same binary ranking loss on the retrieval targets to coordinate the logits range between retrieval and ranking. Specifically, RRA picks the point-wise logits of the target shows from the retrieval outcomes on the **impression** samples, i.e., $\hat{s}^i_{ctr}$ and $\hat{s}^i_{dwm}$, and then computes the same ranking loss of Eqn.1 and Eqn.2 on the selected entries of the target items, where the **impressions with clicks** are regarded as positive and the other **impressions** are regarded as negative. We use $\mathcal{L}^{ctr}_{mask}$ and $\mathcal{L}^{dwm}_{mask}$ to denote the masked binary CTR and DWM losses. It is evident that the masked binary loss aligns the retrieval objectives and training distribution with the ranking objectives, thereby helping the retrieval model further distinguish the clicked videos from the impressed ones (hard negative).

4.3.4 Knowledge distillation loss for retrieval. To directly align the predictions between the retrieval and ranking stages, RRA applies the cross-entropy distillation [8] between the retrieval and ranking predictions on the **impression** samples for transferring knowledge from the ranking model to retrieval model, which can be formulated as:

$$\mathcal{L}^{ctr}_{kd} = -\sigma(s^i_{ctr}) \log \sigma(\hat{s}^i_{ctr}) - (1 - \sigma(s^i_{ctr})) \log(1 - \sigma(\hat{s}^i_{ctr})), \tag{5}$$

$$\mathcal{L}^{dwm}_{kd} = -\sigma(s^i_{dwm}) \log \sigma(\hat{s}^i_{dwm}) - (1-\sigma(s^i_{dwm})) \log(1-\sigma(\hat{s}^i_{dwm})). \tag{6}$$

The distillation loss fills the gap of model capacity, training distribution, and objective, encouraging the retrieval model to learn the complex features and interactions from the ranking model.

4.4 Model Serving in Production Environment

During online serving, RRA would perform a linear fusion on the prediction scores of CTR and DWM with the hyperparameter $0 < w_{serve} < 1$:

$$\hat{s}_{serve} = w_{serve}\hat{s}_{ctr} + (1 - w_{serve})\hat{s}_{dwm}. \tag{7}$$

As the candidate set is limited in the long-video streaming platform, RRA computes the $\hat{s}_{serve}$ based on the efficient two-tower architecture, where the user embeddings are only computed once for each request despite the number of candidates, thereby the computation overhead is much smaller compared to the ranking stage. For the scenarios with a large number of candidates such as Youtube and Tiktok, RRA would perform linear fusion on the user embeddings generated by the CTR and DWM tasks:

$$u_{serve} = w_{serve}u_{ctr} + (1 - w_{serve})u_{dwm} \tag{8}$$

Then, RRA will use u_{serve} to retrieve the candidates from the index built upon the item embeddings through the approximate nearest neighbor methods [9–11].

4.5 Discussion

RRA uses the knowledge distillation method to align the ranking and retrieval stages through synchronously trained teacher and student models (online distillation). The existing methods also introduce two alternative distillation-based training schemes that either adopt the pre-trained model as the teacher (offline distillation) or directly use the pseudo labels of the ranking prediction scores logged from the online service to guide the student training [6]. We will discuss those two methods in this section.

4.5.1 Pre-trained Teacher (Offline Distillation). Most of distillation methods [5, 8, 20, 23, 25, 33] are trained in an offline scheme, where knowledge is transferred from a pre-trained teacher to a student. Offline distillation is simple and easy to be implemented in the RRA framework. We tried this method but the result was slightly worse than the proposed method. The detail could be found in Tab.1.

4.5.2 Pseudo Labeling. Logging the real-time prediction scores of the ranking model as pseudo labels [13] for the student model is a more engineering-efficient training scheme. The main advantage of the pseudo labeling is that the ranking model does not need to be loaded in the GPU memory and compute the ranking scores during the retrieval training, which could save the computation resources and be easily implemented as the ordinary retrieval model. However, in the actual production environment of the long-video recommender system, the ranking model is usually retrained or updated every few hours or even minutes, and the distribution of prediction scores from the different snapshots can be divergent. As the retrieval training set often contains samples across several months, the difference in the distribution of ranking scores will make it hard for the retrieval model to learn stably via knowledge distillation, leading to inferior performance compared to the synchronous teacher in RRA. We also tested adding the snapshot ID as the additional feature to discriminate the distribution difference and adaptive tuning the model according to the predictions from different snapshot IDs, but still failed to achieve comparable performance as online distillation.

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Dataset. As we did not find a public real-world streaming dataset with impression loggings and similar business logic as the considered scenario, we collect three-month impression loggings from the unfamiliar collections of "TV For You" and "Movies For You" in Hulu, one of the largest video streaming platform in the world, for a thorough evaluation. The "TV For You" dataset involves 200M downsampled impressions of roughly 45M clicks while the "Movies For You" dataset involves 120M impressions and 31M clicks.

Table 1: The offline evaluation on the "TV For You" and "Movies For You" collections in the Hulu Dataset.

Model	TV For You			Movies For You		
	Recall@100	WRecall@100	NKTD@300	Recall@100	WRecall@100	NKTD@300
BaseModel	0.7753	0.7400	0.4380	0.8958	0.8418	0.4872
BaseModel + DMTL	0.7866	0.7612	0.4252	0.9012	0.8530	0.4658
BaseModel + parameter sharing	0.8007	0.7640	0.4203	0.9058	0.8561	0.4677
BaseModel + RRA (Offline Distillation)	0.8100	0.7714	0.4146	0.9077	0.8695	0.4487
BaseModel + RRA	0.8125	0.7759	0.4177	0.9156	0.8745	0.4490
ProdModel	0.8284	0.7743	0.4120	0.9134	0.8670	0.4529
ProdModel + DMTL	0.8308	0.7843	0.4155	0.9183	0.8737	0.4336
ProdModel + parameter sharing	0.8347	0.7927	0.4023	0.9290	0.8760	0.4360
ProdModel + RRA (Offline Distillation)	0.8666	0.7992	0.3897	0.9302	0.8790	0.4258
ProdModel + RRA	**0.8680**	**0.8005**	**0.3892**	**0.9325**	**0.8802**	**0.4230**

The last-day samples are used for testing while the others are employed for training.

5.1.2 Metrics. Recall@K measures the ratio of clicked items that occurred in the top-K list for the clicked samples, which is widespread for the retrieval evaluation owing to its simplicity. To evaluate the effect of DWM, we propose Weighted Recall@K (WRecall@K for short) that additionally weights each sample by its corresponding DWM. Moreover, we also adopt the Normalized Kendall Tau Distance (NKTD@N for short) to measure the alignment between the ranking model and the retrieval model, which computes the normalized pairwise disagreements between the ordered recommendation lists generated by the two models:

$$NKTD@N = \frac{2}{N(N-1)} \sum_{\{i,j\} \in I, i<j} \bar{K}_{i,j}(\tau_{retrieval}, \tau_{ranking}), \quad (9)$$

where $\tau_{retrieval}$ and $\tau_{ranking}$ are the recommendation lists given by retrieval and ranking models. $\bar{K}_{i,j}(\tau_{retrieval}, \tau_{ranking})$ measures the retrieval and ranking alignment on the shows i and j. Specifically, if shows i and j are in the same order in $\tau_{retrieval}$ and $\tau_{ranking}$, $\bar{K}_{i,j}(\tau_{retrieval}, \tau_{ranking})$ will be set to 0; otherwise, it will be set to 1. Therefore, the ideal ordering alignment would be achieved when NKTD@N = 0. We select Top-N items generated by the retrieval model as the support set, where N is usually set to a multiple of the number of recalls that can better reflect the improvement brought from the retrieval and ranking consistency in the online production environment.

We adopt Recall@100 and WRecall@100 to measure the retrieving accuracy and NKTD@300 to measure the consistency between two recommended lists.

5.1.3 Baseline method. We compare RRA with another ranking-as-teacher method DMTL [34] and list the main differences below:

- DMTL isolates the model of retrieval and ranking.
- DMTL samples the negative items from the entire corpus for both teacher and student.
- DMTL directly uses target show as input and models the task as a binary classification problem.

5.1.4 Implementation details. To ensure RRA is capable of different models, we propose two variants of retrieval architecture denoted as BaseModel and ProdModel. The structure of BaseModel is similar to YoutubeDNN [4]. For the behavior features and other demographic features, a multi-layer perceptron is used for nonlinear transformation after the embedding layer, without introducing structures such as the self-attention layers and PNN. ProdModel refers to the retrieval structure mentioned in Sec.4 and is more complex than the BaseModel. The Adam optimizer [12] with a constant learning rate of 0.005 and batch size of 2048 is employed for training both models. To ensure fairness, we employ the same negative sampling strategy for all comparisons.

5.2 Offline Comparison

As shown in Tab.1, we compare RRA with DMTL and parameter sharing (retrieval model shares the embedding layers and bottom layers ranking model) variants on both BaseModel and ProdModel. It can be observed that:

- Changing the model structure of BaseModel to ProdModel will bring better retrieval performance, indicating the use of self-attention and PNN enhances the representative ability of the model so that it can learn more information from the input features.
- For both BaseModel and ProdModel, the metrics can be greatly improved by sharing embedding and bottom layers with the ranking model, as most of the parameters lie in the shared bottom layers and thus essentially affect the model performance. In the case of parameter sharing between the two models, the complex feature intersection in the ranking structure and the additional signal can also make the retrieval model learn a more accurate partial order relationship.
- DMTL employs the knowledge distillation from the ranking model to the retrieval model. However, it is still inferior to the "parameter sharing" variant, which indicates the strength of knowledge transfer through the sharing of parameters between ranking and retrieval models. Moreover, as DMTL employs the same training distribution between the two stages while neglecting the sample selection difference, it results in an inferior NKTD and thus could give rise to sub-optimal retrieval performance during online serving.

Table 2: Ablation results of adopting various loss constructors in RRA on the "TV For You" dataset.

Loss Constructors	Recall@100	WRecall@100
$\mathcal{L}_{cls}^{ctr}$	0.8307	0.7707
$\mathcal{L}_{cls}^{*} = \mathcal{L}_{cls}^{ctr} + \mathcal{L}_{cls}^{dwm}$	0.8284	0.7743
$\mathcal{L}_{cls}^{ctr} + \mathcal{L}_{rank}^{ctr}$	0.8370	0.7751
$\mathcal{L}_{mask}^{*}$	0.7906	0.7476
$\mathcal{L}_{kd}^{*}$	0.7983	0.7512
$\mathcal{L}_{cls}^{*} + \mathcal{L}_{rank}^{*}$	0.8347	0.7927
$\mathcal{L}_{cls}^{*} + \mathcal{L}_{rank}^{*} + \mathcal{L}_{mask}^{*}$	0.8401	0.7975
$\mathcal{L}_{cls}^{*} + \mathcal{L}_{rank}^{*} + \mathcal{L}_{mask}^{*} + \mathcal{L}_{kd}^{*}$	**0.8680**	**0.8005**

Table 3: Ablation results of adopting different sampling strategies in RRA on the "TV For You" dataset.

Sample Weight	Recall@100	WRecall@100
$w_{ctr} = 0$	0.8663	0.7980
$w_{ctr} = 1$	0.8666	0.7992
$w_{ctr} = 0.5$	**0.8680**	**0.8005**

- Owing to the design of the partially-shared architectures and the hybrid loss of multi-class classification, masked binary, and distillation, RRA significantly outperforms the comparisons across various scenarios and metrics, indicating the superiority of aligning the training distribution, objective and model capacity between retrieval and ranking stages. Moreover, a simple BaseModel with RRA training can be on par with the ProdModel without RRA training and even perform better on the "Movies For You" collection over both CTR and DWM metrics, which repeatedly verifies the strength of the proposed method. It is also worth mentioning that RRA does not introduce any online serving overhead as the metric lift mainly comes from the additional optimization objectives of multi-class classification, masked binary, and distillation losses. Therefore, RRA can be adopted as a plug-and-play component in any retrieval model.
- In addition to obtaining better recommendation accuracy, RRA also greatly improves the NKTD metrics, indicating that the retrieval outcomes are more consistent with the ranking predictions on the top items. The results are mainly attributable to the design of masked binary loss and the knowledge distillation loss that explicitly fill the gap of the objectives between the two stages.

5.3 Ablation Study

In this section, we disassemble the RRA pipeline and study the ablation effect of each sub-module.

5.3.1 Loss constructors. To study the ablation effect of the various losses in Sec.4.3, we only employ the subsets of losses in RRA and show the performance of the variants in Tab.2. We use $\mathcal{L}^{*}$ to denote both $\mathcal{L}^{ctr}$ and $\mathcal{L}^{dwm}$. From the ablation results, we have the following findings:

- As the weighted softmax cross-entropy loss $\mathcal{L}_{cls}^{ctr}$ assigns different weights to easy negatives, hard negatives, and clicked items, it could already discriminate the variance among those candidates and thus achieve comparable performance with RRA. Based on the CTR loss, the combination of DWM loss would bring a Recall drop but a WRecall lift. The result is in line with the expectation. As the CTR and DWM optimizations are proposed for optimizing discrepant metrics of recall and weighted recall respectively, the linear fusion in Eqn.7 would apparently take the middle point between the two objectives.
- $\mathcal{L}_{rank}^{ctr}$ introduces a positive effect on the retrieval CTR loss $\mathcal{L}_{cls}^{ctr}$ through the parameter sharing and implicit knowledge transfer, which once again proves the benefit of training retrieval model together with the ranking model.
- Retrieval model achieves sub-optimal performance when only trained by masked binary losses or knowledge distillation losses due to the distribution mismatch between the training and inference stages. As described in Sec.1, the retrieval stage needs to find the relevant items matching user interests from the entire corpus. However, the masked binary loss and knowledge distillation loss employ the impression loggings during training, which are mainly hard negative. Due to the training and serving distribution mismatch, the retrieval model is not capable to identify the easy negatives (random negatives) from the hard negatives, and thus performs poorly on the retrieval metrics.
- Based on the multi-class classification loss, the ranking loss, masked binary loss, and knowledge distillation loss gradually improve the retrieval performance owing to the alignment between the retrieval and ranking models that help distinguish the click, hard negative, and easy negative samples. Note that knowledge distillation loss largely improves both metrics compared to the masked binary loss, indicating the significance of mitigating the model capacity discrepancy between the models, and the predominance of distillation-based methods on the knowledge transfer in the recommender system.

5.3.2 Sample weight in multi-class classification. As mentioned in Sec.4.3, RRA assigns different CTR weights w_{ctr} to the samples in the multi-class classification loss of Eqn.3. To study the effect of w_{ctr}, we propose three sample weights by setting w_{ctr} to 0, 0.5 (the best midpoint through tuning), and 1, which correspond to different sampling strategies. From Tab.3, it is obvious that assigning the impression $w_{ctr} = 0.5$ achieves the best performance, reflecting the importance to discriminate the samples of different types.

5.3.3 Negative sampling in multi-class classification. We also compare different negative sampling methods described in the previous works with the proposed All Negative (AN), including Random Negative (RN) [31], Batch Negative (BN) [4], Batch Negative with popularity Correction (BNC) [32], Hard Negative (HN) [9], and Mixed Negative (MN) [30] that combines random negative (all negative) and hard negative. As illustrated in Tab.4, it is obvious that

Table 4: Ablation results of adopting different negative sampling strategies in RRA on the "TV For You" dataset. RN n selects n random negative samples from the entire corpus. BN and BNC denote the batch negative sampling w/ and w/o popularity correction. HN $r_{\text{low}} - r_{\text{up}}$ selects the items with rank positions between r_{low} and r_{up} as negative. AN represents the proposed all-negative sampling method. MN $r_{\text{low}} - r_{\text{up}}$ means combining AN with HN $r_{\text{low}} - r_{\text{up}}$.

Negative Sampling	Recall@100	WRecall@100
RN 20	0.7584	0.7066
RN 100	0.7813	0.7403
RN 1000	0.8467	0.7895
BN	0.8444	0.7913
BNC	0.8474	0.7961
HN 100-200	0.7361	0.6987
HN 300-500	0.7724	0.7264
AN	**0.8680**	**0.8005**
MN 100-200	0.8575	0.7943
MN 300-500	0.8665	0.7979

the retrieval performance improves along with the increase of the negative samples, indicating that the sampling rate should be as low as possible for training the retrieval model. Popularity correction can improve the batch negative sampling, but it is still inferior to the proposed all-negative strategy. Although mixed negative combines additional hard negative samples with the random negatives, it can not bring further improvement over the adopted method that obtained the best performance. Additionally, the model trained only by hard negatives is significantly worse than the others as those negative samples bias towards the hard cases which mismatches the serving distribution where the majority of target shows are easy negatives.

5.4 Online Evaluation

To evaluate the performance of RRA on the real-world long-video streaming platform, we conduct online A/B experiments on the unfamiliar "TV For You" and "Movies For You" collections of Hulu. The Hulu recommender system employs the widespread multi-channel retrieval [15], where candidates are generated by multiple engines capturing different dimensions to characterize various user purposes. We replace the personalized retrieval channel that employs the CTR-optimized BaseModel with ProdModel + RRA and select the best-tuned $w_{serving}$ through grid search according to the offline experimental metric. The experiment results show that ProdModel + RRA outperforms the baseline by 2%/3% on Local consumptions (playback within collections) and 7%/4% on DWM for "TV For You"/"Movies For You" collections, representing that RRA not only brings more instant viewing time within the unfamiliar collections but also leads to more subsequent behaviors in the familiar collections, verifying the importance of optimizing retrieval model towards the long-term objective. Meanwhile, the order returned by RRA is more consistent with the online ranking model, i.e., 8%/6% increase of the item exposure across multiple channels and a 5%/3% decrease on NKTD@300 over "TV For You"/"Movies For You" collections, indicating that RRA better aligns retrieval and ranking stages. Based on the DWM objective and alignment between retrieval and ranking stages, RRA brings about a significant 0.74% increase in the platform-level viewing time, which is prominent for the unfamiliar recommendation in the long-video streaming platform.

6 CONCLUSION

In this paper, we propose RRA for the unfamiliar recommendation in the long-video streaming platform. A Discovery-induced Watching Minutes (DWM) is proposed as the additional objective to capture users' long-term engagement in the entire platform derived from the initial playback in the unfamiliar collections. To mitigate the discrepancy between retrieval and ranking, RRA coordinates the retrieval and ranking models via a partially-shared architecture, then proposes the various losses during synchronous training, including the multi-class classification loss, masked binary loss, and knowledge distillation. The broad offline experiments exhibit that RRA outperforms the compared methods across various scenarios and architectures without introducing any serving overhead, indicating the efficiency, effectiveness, and universality of the proposed method. Moreover, RRA brings a significant lift over the platform-level viewing time during online evaluation, indicating the importance of using DWM and aligning the retrieval and ranking stages in the unfamiliar recommendation of the long-video streaming platform. RRA has been deployed in the "TV For You" and the "Movies For You" collections of Hulu and served tens of millions of subscribers.

REFERENCES

[1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[2] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 456–464.

[3] 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 *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 7–10.

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

[5] Jack FitzGerald, Shankar Ananthakrishnan, Konstantine Arkoudas, Davide Bernardi, Abhishek Bhagia, Claudio Delli Bovi, Jin Cao, Rakesh Chada, Amit Chauhan, Luoxin Chen, et al. 2022. Alexa teacher model: Pretraining and distilling multi-billion-parameter encoders for natural language understanding systems. (2022).

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

[7] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[8] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[9] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[10] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010), 117–128.

[11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[12] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[13] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 896.

[14] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[15] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. In *SIGIR*. 1493–1502.

[16] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3181–3189.

[17] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[18] Alessandro Magnani, Feng Liu, Suthee Chaidaroon, Sachin Yadav, Praveen Reddy Suram, Ajit Puthenputhussery, Sijie Chen, Min Xie, Anirudh Kashi, Tony Lee, et al. 2022. Semantic Retrieval at Walmart. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3495–3503.

[19] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. PinnerFormer: Sequence Modeling for User Representation at Pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3702–3712.

[20] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. 2020. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2339–2348.

[21] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. 2022. RankFlow: Joint Optimization of Multi-Stage Cascade Ranking Systems as Flows. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 814–824.

[22] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[24] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 269–278.

[25] Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2289–2298.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[27] Zhe Wang, Liqin Zhao, Biye Jiang, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2020. Cold: Towards the next generation of pre-ranking system. *arXiv preprint arXiv:2007.16122* (2020).

[28] Xuyang Wu, Alessandro Magnani, Suthee Chaidaroon, Ajit Puthenputhussery, Ciya Liao, and Yi Fang. 2022. A Multi-task Learning Framework for Product Ranking with BERT. In *Proceedings of the ACM Web Conference 2022*. 493–501.

[29] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged features distillation at taobao recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2590–2598.

[30] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H. Chi. 2020. Mixed Negative Sampling for Learning Two-tower Neural Networks in Recommendations. In *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Amal El Fallah Seghrouchni, Gita Sukthankar, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 441–447.

[31] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[32] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[33] Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1285–1294.

[34] Zhong Zhao, Yanmei Fu, Hanming Liang, Li Ma, Guangyao Zhao, and Hongwei Jiang. 2021. Distillation based Multi-task Learning: A Candidate Generation Model for Improving Reading Duration. *arXiv preprint arXiv:2102.07142* (2021).

[35] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: A multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, 43–51.

[36] 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.

[37] 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*. ACM, 1059–1068.

[38] 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.