

Personalized Transformer-based Ranking for e-Commerce at Yandex

Kirill Khrylchenko
Yandex
Moscow, Russia

Alexander Fritzler
Yandex
Almaty, Kazakhstan

ABSTRACT

Personalizing user experience with high-quality recommendations based on user activity is vital for e-commerce platforms. This is particularly important in scenarios where the user's intent is not explicit, such as on the homepage. Recently, personalized embedding-based systems have significantly improved the quality of recommendations and search in the e-commerce domain. However, most of these works focus on enhancing the retrieval stage.

In this paper, we demonstrate that features produced by retrieval-focused deep learning models are sub-optimal for ranking stage in e-commerce recommendations. To address this issue, we propose a two-stage training process that fine-tunes two-tower models to achieve optimal ranking performance. We provide a detailed description of our transformer-based two-tower model architecture, which is specifically designed for personalization in e-commerce.

Additionally, we introduce a novel technique for debiasing context in offline models and report significant improvements in ranking performance when using web-search queries for e-commerce recommendations. Our model has been successfully deployed at Yandex, serves millions of users daily, and has delivered strong performance in online A/B testing.

CCS CONCEPTS

• **Information systems** → **Information retrieval; Learning to rank; Personalization; Recommender systems**; • **Computing methodologies** → **Learning from implicit feedback; Neural networks**.

KEYWORDS

deep learning, personalization, recommender systems, e-commerce, learning-to-rank, debiasing

ACM Reference Format:

Kirill Khrylchenko and Alexander Fritzler. 2023. Personalized Transformer-based Ranking for e-Commerce at Yandex. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/XXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXX.XXXXXXX>

1 INTRODUCTION

E-commerce platforms have become an essential part of our society, allowing us to exploit and explore our interests through access to a wide variety of items. Recommender systems aim to alleviate information overload and present relevant and interesting results to users from billions of potential choices.

Recently, many large e-commerce platforms (e.g. Amazon [1, 28], Taobao [6, 21, 59], eBay [45, 56], Walmart [25, 26, 46], JD.com [44, 52], Facebook Marketplace [14, 23]) have integrated deep learning into their recommender systems and search engines. However, most of these works focus on the retrieval stage, which produces a small set of candidates for further filtering and ranking by subsequent stages. Similarity features produced by embedding-based retrieval (EBR) models can also improve ranker models. For example, Liu et al. [23] report +1.24% online in-conversation uplift in Que2Search from using two-tower cosine similarity feature trained with retrieval loss.

Although some models attempt to bridge the gap between retrieval and ranking [14, 59, 59], we argue that retrieval and ranking still vary significantly. Retrieval is typically concerned with recall optimization and heavily utilizes weakly supervised signals like clicks. Ranking is based on sparser signals, such as purchases.

In this paper, we introduce a ranking-oriented transformer-based two-tower model for personalization. The user-item similarity features generated with our approach greatly improve the quality of subsequent ranking models. Our key contributions are as follows:

- We present an inductive transformer-based two-tower model architecture that is specifically optimized for the personalized ranking of e-commerce recommendations. To the best of our knowledge, this is the first work to focus on ranking-oriented personalization transformers for recommendations in the e-commerce domain.
- We propose a novel two-stage training technique that combines recall-oriented pre-training with ranking-oriented fine-tuning. This approach results in significant improvements in ranking quality and addresses the challenge of sparse positive ranking signals.
- We show that integrating web-search history into e-commerce recommendations significantly improves ranking quality.
- We define the concept of context bias and propose a context debiasing technique to remove context-related signal from offline, non real-time models.
- Our model has been successfully deployed in an industrial setting and has demonstrated strong performance improvements on the homepage and cart landing page of the large e-commerce platform Yandex Market through online A/B testing.

2 RELATED WORKS

To our knowledge, the first two-tower model was proposed by Microsoft: DSSM [16] generated query and document embeddings for the web-search domain. Many companies have since adopted two-tower models for various information retrieval tasks such as recommendations, search, and ads. Information retrieval involves a multi-stage process of information filtering: billion-sized catalogs are filtered down to a few candidates to be presented to the user. The first stage, called retrieval, is where two-tower models are most commonly used. EBR systems usually constitute one of multiple retrieval channels [23, 26, 28, 59]. Another popular choice is the inverted index with boolean matching based on text or visual signals [4, 42, 57, 58]. While inverted indices have good relevance due to exact term matching, EBR is tolerant to misspellings and allows for personalization and context awareness. EBR models are typically trained to optimize softmax loss with in-batch or uniformly sampled negatives. Yi et al. [51] used importance sampling to correct sampling bias of in-batch negatives for Youtube. Yang et al. [50] introduced further improvements for retrieval using mixed negative sampling for Google App Store. Recently, there has been a trend to increase consistency between the ranking and retrieval stages with auxiliary losses [14, 44, 59].

The **ranking stage** is usually one of the last stages in information filtering, where compute-intensive models are used on hundreds or thousands of candidates. Deep learning models for personalized ranking mostly follow the Embedding&MLP paradigm proposed by Cheng et al. [7]. **Such models tend to abandon two-tower architecture in favor of fusing user, context, and target item information.** YoutubeDNN [8] formed user video-watching and searching activity vectors by averaging historical user event embeddings. DIN [61] used a convex combination of historical user event embeddings, essentially an attention layer with the target item as query and historical events as keys and values. DIEN [60] and BST [6] improved on DIN with GRU and transformers. However, **these models use an early fusion of the target item and user history items, which is very costly in production even for the re-ranking stage and limits our ability to use batch serving.** This motivates us to explore two-tower models. There are works that employ transformer-based two-tower models to produce similarity-based features for ranking in search [22, 24] and ads [53]. However, these models are not personalized and only use query and document information.

Personalization with sequential recommenders based on user histories is largely associated with the next-item-prediction (NIP) task, which essentially turns recommendation into a language modeling problem. First works in this field were motivated by the progress in natural language processing, e.g. ELMo [31], Transformer [43], GPT [33], BERT [10]. GRU4Rec [15], Caser [41] and SASRec [19] were the pioneer works to model the NIP task with GRU, CNNs, and transformers respectively. Bert4Rec [40] trained a recommender in a BERT-like bidirectional fashion with MLM. CL4Rec [49] used contrastive learning for pre-training, and CARCA [34] promoted inductive context-aware models with content-based item embeddings. However, most of these works conduct evaluation on academic datasets with questionable practices such as sampled negatives [20] and absence of a time-based train-test split [17, 39]. Furthermore, the **NIP task is subject to selection bias which encourages models**

to imitate the logging policy [9]. Additionally, due to the scale of academic datasets, these works are prone to unnecessary or even harmful **inductive bias: customized model architectures, auxiliary losses, and special pre-training regimes that do not benefit real-world web-scale recommendations.** Large companies also conduct research and develop sequential recommenders: Pinterest [29], Alibaba [6, 21], eBay [38, 45, 56], NAVER [36, 37], Etsy [2], Spotify [11], etc. **They usually provide correct evaluation schemes, use web-scale datasets, and sometimes provide results of online A/B testing.**

To combat the feedback loop, it's **important to mitigate the effects of various recommendation biases like position, selection, examination, trust, and popularity bias.** There are several debiasing approaches: (1) reweighting samples based on inversed propensity scores [5, 18]; and (2) modeling bias explicitly with a separate model [12, 13, 54, 55], detaching the bias model during deployment. Motivated by the second approach, we introduce the concept of context bias and propose a context debiasing technique to improve the ranking quality of offline models.

Although enriching user e-commerce history with web-search queries is a case of cross-domain recommendations (CDR), we do not delve into CDR research because it is not a goal of our work to explore and improve upon various CDR methods. To the best of our knowledge, the most notable works that improve the quality of e-commerce recommendations with web-search data are from NAVER [36, 37].

3 MODELING

In this section, we describe our two-tower model architecture, shown in Figure 1. We discuss the item tower, user tower, and similarity function separately, before introducing an additional context tower for context debiasing. Finally, we present our two-stage training regime, which involves pre-training the model for retrieval-oriented tasks and then fine-tuning it for ranking-focused tasks.

3.1 Item Tower

In the e-commerce domain we are tasked with recommending products to the users. Unlike some domains such as web-search, products provide a rich source of information including category, title, description, structured attributes (e.g. brand, color, flavor, material), images, and reviews. Product catalogs are very dynamic with some items available in a single quantity and new items arriving daily, making it **important to consider the cold-start and distribution shift problems.** To increase generalization, avoid memorization and alleviate cold-start we do not use identifier-based learnable embeddings. Instead, **we employ content-based item representations using only titles.**

Product titles are processed as CBOW [27] using a wordpiece [47] tokenizer built on web-search data with a vocabulary of 103295 tokens. **Token embeddings are summed which does not degrade performance compared to averaging.**

The item tower architecture was motivated by our experience with personalization transformer-based models for web search and display advertising. Initially, in web-search we used a cross-encoder model with user and item features fused in the same transformer encoder. Later on we switched to two-tower models to handle

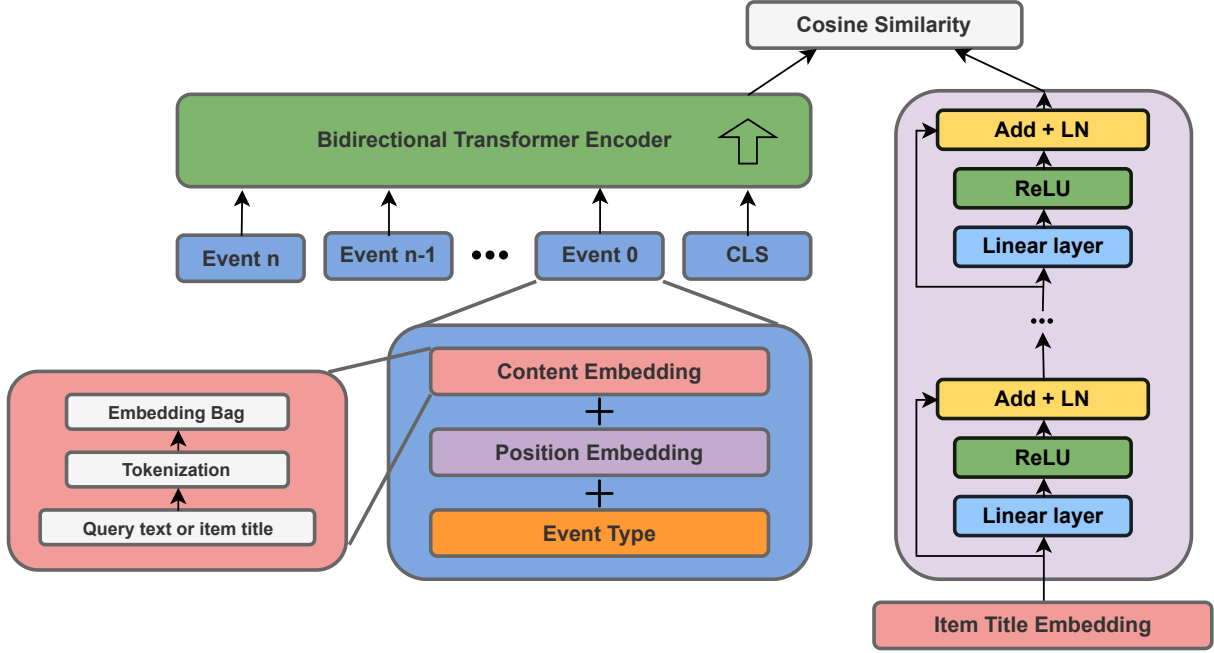


Figure 1: Model architecture. Add+ln denotes $\text{layernorm}(x + y)$.

millions of RPS in display advertising. The quality of the model decreased significantly when we used a Multi-Layer Perceptron (MLP) as an item tower. However, **using transformer encoder as an item tower on top of products represented as a single CBOW title embedding produced good results.**

Simplifying the transformer encoder with a single input embedding lead to the architecture presented on the right in figure 1. This architecture differs from the MLP in that it includes residual connections and layer normalizations, similar to those found in a transformer:

$$x_{k+1} = \text{LayerNorm}_k(\text{ReLU}(\text{Linear}_k(x_k)) + x_k)$$

where k is a layer number and $x_1 \in \mathbb{R}^d$ is an input title embedding.

With such architecture, we were able to preserve most of the initial cross-encoder quality gains. Incorporating contrastive learning for pre-training further improved the model quality compared to the cross-encoder.

Following the item tower, the item embedding is l_2 normalized to unit length.

3.2 User Tower

To our knowledge, when choosing which types of user events to use in a sequence encoded by a transformer, **the first best guess is to use the same types of events you are trying to predict. We form a chronologically ordered sequence from the user's clicks, add-to-carts, add-to-favorites, and purchases.**

We also observed quality gains from enriching user histories with web-search queries. Interestingly, using web-search document clicks did not lead to any further improvements. We explore two

ways of incorporating web-search history: using a separate transformer encoder on a fixed number of the most recent user's web-search queries and fusing web-search information with e-commerce activity into a single event sequence. Moreover, we believe that **combining diverse data types, such as queries and products, yields even better results in an inductive setting that utilizes content-based event representations.**

Three types of embeddings are summed up to form an event embedding:

- **Content embedding.** Queries and product titles are tokenized using the wordpiece tokenizer from section 3.1. **The embedding matrix is shared across queries and products.**
- **Positional embedding.** Each event is assigned a chronologically reversed absolute position: the latest event gets position 0, the second latest gets position 1, and so on. **Position embeddings are learnable.**
- **Event type embedding.** Every event type, including click, add-to-cart, add-to-favorites, purchase, web-search query, **has a learnable embedding.**

Together with learnable [CLS] embedding, event embeddings are fed to a bi-directional transformer encoder. A post-transformer contextualized [CLS] embedding is l_2 normalized and used as a final user representation. We use a standard transformer encoder architecture [10, 43] with post-normalization. All embeddings, except for [CLS], are layer-normalized prior to the transformer.

Similar to Pinnerformer [29], we opt to use a single user embedding to simplify maintenance and deployment. Our model is used in a batch serving scenario with user and item embeddings being recalculated daily. Multiple user embeddings would result in a

linearly scaled memory cost for KV-storage and complicate integration into downstream applications. However, using multiple user embeddings usually produces better quality and helps to enforce diversity, which is why we usually employ multiple embeddings for real-time models.

We use a bi-directional transformer encoder without causal masking and teacher forcing, only the [CLS] representation is used to produce scores.

3.3 Similarity function

We use inner product as a user-item similarity function. Since we l_2 normalize embeddings, it is equivalent to cosine similarity:

$$r_{ui} = \frac{\langle v_u, v_i \rangle}{\|v_u\| \|v_i\|}$$

where $v_u, v_i \in \mathbb{R}^d$ are user u and item i embeddings. In our experiments, we found that normalizing embeddings usually results in a more stable training process.

3.4 Context Debiasing

When a recommendation request comes, typically a variety of contextual data, such as **device, operational system, time of day, seed item, search query, is available along with user and item information**. Using context information in real-time models is paramount and usually brings significant improvements.

We will discuss deployment in more detail in Section 4.4, but it's worth noting that we **deploy our model in batch-serving mode, i.e. user and item embeddings are recalculated on a daily basis. As a result, we do not have access to context information during model inference**. Since the context has very useful signal, our models tend to implicitly infer current user context from history. However, it is unnecessary to spend model and user embedding capacity for implicit context information. We expect that our gradient boosting trees ranker model (e.g. Catboost [32]), for which we are constructing a similarity-based feature, already utilizes context information efficiently. Additionally, our data is skewed towards certain user contexts that are more prevalent in real-world scenarios, which can impact our performance for less popular contexts.

Inspired by debiasing techniques aimed at mitigating effects from various biases, we introduce context debiasing technique for alleviating redundancy of context information in our ranker. Without context debiasing, our model estimates the following probability:

$$P(i|u) = \sum_{ctx \in C} P(i|u, ctx)P(ctx|u)$$

where $P(i|u)$ is a probability that the user u clicks on item i , and C is a set of all possible contexts.

Instead, during training we learn a separate context tower using actual user context:

$$P(i|u, ctx) = \sigma(r_{ui} + r_{ctx})$$

where r_{ui} is learned similarity between user u and item i , and r_{ctx} is a learnable scalar based on a given context ctx .

During deployment, we detach the context tower and use only user-item similarity. A possible modification to this technique would be to fuse context with item and user, for example, by forming a context embedding and calculating the inner product between

context and item. However, in our experiments, we didn't see any improvements with this approach.

Similar to Zhang et al. [54], we attempted to apply dropout to the output of the context tower but it did not lead to any quality improvements. This could be related to the fact that our context tower has the simplest possible form. We do not exclude the possibility that dropout will be useful with a more complex context tower.

3.5 Two-stage training

The most common loss function for deep learning two-tower personalization models is a sampled softmax loss with random and in-batch negatives, which essentially makes the model retrieval-focused. Negative implicit feedback based on impressions is rarely used, because: (1) using logs with non-positive impressions dramatically increases pipeline complexity; (2) such models perform poorly on retrieval tasks due to sample selection bias; (3) all impressed items are very similar [45] and examination bias leading to very noisy data with large amounts of false negatives.

However, the sample selection bias problem goes both ways. When trained with sampled negatives, models underperform on ranking scenarios which present much harder impressed negatives. We demonstrate that tuning the model directly on a ranking scenario with pairwise ranking loss and impressed negatives results in significant gains.

Still, due to the data sparsity of positive feedback (such as purchases), training large two-tower models on ranking from scratch yields almost zero results. This issue can be mitigated with transfer learning. We propose a two-stage training scheme for training ranking-oriented deep learning two-tower models.

3.5.1 Pre-training stage. In the pre-training stage, the model is trained in a standard retrieval-oriented regime with sampled softmax loss function:

$$\mathcal{L}_{pretrain}(u, p, N) = -\log \frac{\exp(\tau \cdot r_{up})}{\exp(\tau \cdot r_{up}) + \sum_{n \in N} \exp(\tau \cdot r_{un})}$$

where τ represents the temperature parameter, r_{ui} denotes the similarity between user u and item i , p corresponds to a positive item, and N is a collection of in-batch negatives. Along with on-device in-batch negatives, we utilize item embeddings from all other GPU workers as negative samples. **User clicks, add-to-carts, add-to-favorites and purchases are used as positive item interactions.**

3.5.2 Fine-tuning stage. During the fine-tuning stage, we use a **separate loss for each type of the positive impressed signal, including clicks, add-to-cart, add-to-favorites, and purchases**. To ensure that our similarity scores remain well-calibrated during continuous training, we employ a combination of pairwise and pointwise losses.

Initially, we used a separate pointwise loss for each type of the positive signal, but later we **switched to a single pointwise loss based on clicks**:

$$\text{BCE}_{\text{click}}(u, i) = -y_i \log f_{ui} - (1 - y_i) \log(1 - f_{ui})$$

where y_i indicates whether item i is clicked, $f_{ui} := \sigma(\alpha_{cl} r_{ui} + \alpha_{ctx} r_{ctx} + \beta_{cl})$ is the predicted probability of item i being clicked

by user u , and $\alpha_{cl}, \alpha_{ctx}, \beta_{cl}$ are learned scalar parameters. r_{ctx} is the context tower prediction.

The initial pairwise loss for each type of positive signal had a form:

$$\begin{aligned} \text{BPR}_k(u, p, n) &= -\log \sigma(\gamma_k(r_{up} - r_{un})) = \\ &= -\log \frac{e^{\gamma_k r_{up}}}{e^{\gamma_k r_{up}} + e^{\gamma_k r_{un}}} \end{aligned} \quad (1)$$

where k denotes a type of positive feedback, u, p, n represent user, positive and negative items respectively, and γ_k is a learned scalar.

We adopted a solution from Bai et al. [3] to pairwise losses, **replacing the exponents with sigmoids** in equation 1. With this approach, we achieve **well-aligned ranking and regression objectives**. Furthermore, sigmoids allow us to integrate context debiasing into pairwise losses. Our **final pairwise ranking loss** is:

$$\text{BPR}_k(u, p, n) = -\log \frac{f_{up}}{f_{up} + f_{un}}$$

where $f_{ui} := \sigma(\gamma_k r_{ui} + \gamma_{k,ctx} r_{ctx} + \beta_k)$, with additional learned scalars $\gamma_{k,ctx}, \beta_k$.

Our final objective for fine-tuning the model is shown below. For brevity, we omit specific arguments of each loss component:

$$\mathcal{L}_{\text{finetune}} = \text{BPR}_{\text{click}} + \text{BPR}_{\text{cart}} + \text{BPR}_{\text{fort}} + \text{BPR}_{\text{prch}} + \text{BCE}_{\text{click}} \quad (2)$$

where the pairwise loss is calculated for each targetwise ordered item pair within a recommendation request and pointwise loss is calculated for each impressed item.

4 EXPERIMENTS

4.1 Dataset

We gather one year of logs from Yandex e-commerce platform Yandex Market for training, including clicks, add-to-cart, add-to-favorites, purchases, and web queries. For fine-tuning, we collect impressions from all recommendation surfaces at Yandex Market. Due to the proprietary nature of the data, we don't disclose dataset sizes.

For each positive user interaction, a pre-training sample is formed from the user history up to that interaction with a one-day delay. Such delay for history events is important because (1) it is consistent with the production daily batch serving job scenario, and (2) **setting a low history delay may lead to an overly easy pre-training task.** A single item usually generates a sequence of positive interactions within a short time period, e.g. click \rightarrow add-to-cart \rightarrow purchase, which is why it becomes easy to predict add-to-carts based on clicks and predict purchases based on add-to-carts with in-batch negatives and short history delay. Unlike the fine-tuning stage, pre-training also utilizes organic positive interactions produced by the user outside of recommendations.

For fine-tuning, we utilize impressions from all recommendation surfaces on Yandex Market. We group items impressed together and filter out groups without any positive interactions. An impression is considered positive if the impressed item had a positive interaction with the user within a short period following the recommendation. To speed up training and simplify the implementation of pairwise losses, we pack together all impressed items from the same recommendation surface into a single dataset record.

Table 1: Incremental Model Improvements (Relative nDCG Improvement).

	Model	Retargeting	Discovery
1	Sampled Softmax Loss	+0.233%	+0.882%
2	Two-stage Training	+0.476%	+1.377%
3	Calibrated Pairwise Ranking	+0.501%	+1.367%
4	Retaining Single Pointwise Loss	+0.521%	+1.405%
5	History Length (256 to 512)	+0.541%	+1.588%
6	Pointwise Loss Weight (1.0 to 0.1)	+0.566%	+1.631%
7	Context Debiasing	+0.586%	+1.645%
8	Web-Search Queries	+0.677%	+2.061%
9	History Length (512 to 1024)	+0.755%	+1.981%

Although almost every recommendation surface has its own GBT ranker, for brevity we provide results of offline evaluation only for the most user-centric surfaces: retargeting and discovery. Retargeting is an unconstrained form of personalization akin to eBay's Recently Viewed Items [45] module. Discovery limits recommended items to previously undiscovered ones. We use the next ten days after the training period to evaluate our model as a feature in the GBT ranker. As a metric, we report a test relative nDCG gain from including our similarity feature in the GBT feature set.

Context debiasing tower uses recommendation surface identifier and user device as input features. The context tower produces a sum of two learned scalar values, for surface and device respectively.

4.2 Implementation Details

We use quite a large amount of user events — 1024 latest user events. The **transformer encoder has four layers with hidden size 256 and four attention heads.** The candidate tower has four layers with hidden size 1024. **The temperature parameter for softmax loss is learnable.** We split embedding matrices, transformer, candidate tower, and loss parameters into separate parameter groups with groupwise gradient norm clipping and differently tuned learning rates. Learning rate is warmuped for 2500 steps and then linearly decays till the end of the training. Pre-training is done on 16 A100 40g hosts with an effective batch-size of 2048 while **fine-tuning** is done on 8 A100 40g hosts with an **effective batch size of 4096.** Pre-training is done for three epochs and fine-tuning takes a single epoch due to overfitting. **Training is done in a multi-host distributed setting with PyTorch [30] and DeepSpeed [35].**

We continuously fine-tune the model on new chunks of data, weekly. During continuous training, we freeze sigmoid inner parameters (α_k, β_k). Also, we use a constant learning rate for all parameter groups. Only second training stage, presented in equation 2, is employed. Optimizer state is not reused across iterations.

4.3 Offline Experimental Results

The main experimental results are presented in Table 1 as a sequence of incremental improvements. The baseline (L1) is only trained on the retrieval task. Fine-tuning on the pairwise ranking task (L2) brings significant improvements. Calibrated ranking (L3) bridges the gap between pointwise and pairwise losses. Removing all pointwise losses, with the exception of clickwise loss (L4),

Table 2: Comparison of Training Regimes for Two-Stage Training (Relative nDCG Improvement).

Pre-training	Fine-tuning	Retargeting	Discovery
-	✓	+0.031%	+0.033%
✓	-	+0.233%	+0.882%
✓	✓	+0.476%	+1.377%

Table 3: Impact of Enriching User History with Web-Search Queries (Relative nDCG Improvement).

	Fusion	History (ecom + web)	Retargeting	Discovery
1	-	512 + 512	+0.704%	+1.787%
2	✓	512	+0.677%	+2.061%
3	✓	1024	+0.755%	+1.981%

demonstrates an uplift in quality in both retargeting and discovery scenarios. Scaling history length to 512 (L5) greatly improves discovery. We then reduce the importance of pointwise loss (L6) and apply context debiasing (L7). Finally, we enrich user histories with web-search queries. First, we keep the same maximum history length as in previous measurements (L8) and then we improve on retargeting by expanding the history length to 1024 events (L9).

The benefits of two-stage training are illustrated in Table 2. The sparsity of positive feedback in the pairwise ranking scenario renders the one-stage training approach ineffective, whereas pre-training on the retrieval task mitigates this effect.

Enhancing user history with web-search queries works better with an early fusion of e-commerce and web-search data, which is demonstrated in the Table 3. When merging history (L2, L3), we take the chronologically latest user events across both web-search and e-commerce activity. Without fusion (L1), we take 512 latest e-commerce events and 512 latest web-search events and train separate transformer encoders to form two embeddings for each user. Single scalar prediction value is calculated by summing up the scaled inner products for each of the user embeddings with an item embedding. During evaluation, we form two distinct features, separately calculating user-item similarities for web-search and e-commerce user representations. Although comparing single feature to multiple features is unfair, fusing information still works better.

4.4 Online Deployment

User and item embeddings are recalculated daily with a batch serving job. Once recomputed, the user embeddings are uploaded to kv-storage. During serving, the item embeddings are stored in RAM as a hash table and embedding lookup is performed when necessary.

We report the number of billed orders that are attributed to recommendations. If an item has been impressed to a user within a short period before the item's order, the item's order is attributed to recommendations. An order as a whole is attributed to recommendations if at least one item in the order is attributed to recommendations. An order is marked as new if it has items that the user has never encountered on our platform before the recommendation.

Table 4: Online A/B Test Results.

Surface	#RecOrders	#New RecOrders
Homepage	+6%	+10%
Cart page	+5%	+7%

Table 4 shows that our model increased the number of orders attributed to recommendations by 6% on the homepage and by 5% on the cart page. New orders attributed to recommendations increase by +10% on the homepage and by +7% on the cart page. Furthermore, we increased the time spent on the next visited page after the homepage by +1.5%.

Our online A/B test revealed an effect similar to Pinterest [48]: the uplift from our model was substantially higher in the first few days compared to the rest of the A/B test.

5 CONCLUSION

In this paper, we demonstrated the importance of ranking-focused fine-tuning of deep learning embedding-based personalization models. We proposed a two-stage training procedure with retrieval-oriented pre-training and ranking-oriented fine-tuning and introduced context debiasing to improve offline models. Enrichment of user e-commerce histories with web-search data demonstrated further improvements. Lastly, we described in detail our two-tower transformer-based personalization model for e-commerce which was validated through offline and online experiments. The described model has been deployed at the Yandex e-commerce platform Yandex Market to serve main app traffic.

We are currently working on a multi-domain real-time context-aware two-tower transformer-based model for both e-commerce recommendations and search at Yandex Market. We also explore ways to improve item embeddings with (1) item transformers, (2) additional product data like structured attributes, descriptions, and product images, and (3) graph neural networks.

ACKNOWLEDGMENTS

We would like to express our gratitude to the team at Yandex Market for their support during the research and deployment of this model. In particular, we would like to thank Mikhail Denisenko for his significant contribution to this project. We are also grateful to Artur Ilichev and Ivan Lapitsky for their assistance. Additionally, we want to acknowledge the rest of our Recsys team, including Ivan Guschenko-Cheverda, Vsevolod Svetlov, and Sergey Ovcharenko. Without their hard work and dedication, this research would not have been possible.

REFERENCES

- [1] Qingyao Ai, Daniel N. Hill, S. V. N. Vishwanathan, and W. Bruce Croft. 2019. A Zero Attention Model for Personalized Product Search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 379–388. <https://doi.org/10.1145/3357384.3357980>
- [2] Alaa Awad, Denisa Roberts, Eden Dolev, Andrea Heyman, Zahra Ebrahimzadeh, Zoe Weil, Marcin Mejran, Vaibhav Malpani, and Mahir Yavuz. 2023. adSformers: Personalization from Short-Term Sequences and Diversity of Representations in Etsy Ads. arXiv:2302.01255 [cs.LG]

- [3] Aijun Bai, Rolf Jagerman, Zhen Qin, Pratyush Kar, Bing-Rong Lin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Regression Compatible Listwise Objectives for Calibrated Ranking. arXiv:2211.01494 [cs.IR]
- [4] Sean Bell, Yiqun Liu, Sami Alsheikh, Yina Tang, Edward Pizzi, M. Henning, Karun Singh, Omkar Parkhi, and Fedor Borisjuk. 2020. GrokNet: Unified Computer Vision Model Trunk and Embeddings For Commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) (KDD '20). Association for Computing Machinery, New York, NY, USA, 2608–2616. <https://doi.org/10.1145/3394486.3403311>
- [5] Can Chen, Chen Ma, Xi Chen, Sirui Song, Hao Liu, and Xue Liu. 2022. Unbiased Implicit Feedback via Bi-level Optimization. arXiv:2206.00147 [cs.IR]
- [6] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data* (Anchorage, Alaska) (DLP-KDD '19). Association for Computing Machinery, New York, NY, USA, Article 12, 4 pages. <https://doi.org/10.1145/3326937.3341261>
- [7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishu Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) (DLRS 2016). Association for Computing Machinery, New York, NY, USA, 7–10. <https://doi.org/10.1145/2988450.2988454>
- [8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA.
- [9] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Offline Evaluation for Reinforcement Learning-Based Recommendation: A Critical Issue and Some Alternatives. *SIGIR Forum* 56, 2, Article 3 (jan 2023), 14 pages. <https://doi.org/10.1145/3582900.3582905>
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv abs/1810.04805* (2019).
- [11] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) (WWW '22). Association for Computing Machinery, New York, NY, USA, 2036–2047. <https://doi.org/10.1145/3485447.3512077>
- [12] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: A Position-Bias Aware Learning Framework for CTR Prediction in Live Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) (RecSys '19). Association for Computing Machinery, New York, NY, USA, 452–456. <https://doi.org/10.1145/3298689.3347033>
- [13] Malay Haldar, Prashant Ramanathan, Tyler Sax, Mustafa Abdo, Lanbo Zhang, Aamir Mansawala, Shulin Yang, Bradley Turnbull, and Junshuo Liao. 2020. Improving Deep Learning for Airbnb Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) (KDD '20). Association for Computing Machinery, New York, NY, USA, 2822–2830. <https://doi.org/10.1145/3394486.3403333>
- [14] Yunzhong He, Yuxin Tian, Mengjiao Wang, Feier Chen, Licheng Yu, Maolong Tang, Congcong Chen, Ning Zhang, Bin Kuang, and Arul Prakash. 2023. Que2Engage: Embedding-based Retrieval for Relevant and Engaging Products at Facebook Marketplace. *CoRR abs/2302.11052* (2023). <https://doi.org/10.48550/arXiv.2302.11052>
- [15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. <http://arxiv.org/abs/1511.06939> cite arxiv:1511.06939Comment: Camera ready version (17th February, 2016) Affiliation update (29th March, 2016).
- [16] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management* (San Francisco, California, USA) (CIKM '13). Association for Computing Machinery, New York, NY, USA, 2333–2338. <https://doi.org/10.1145/2505515.2505665>
- [17] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A Critical Study on Data Leakage in Recommender System Offline Evaluation. *ACM Trans. Inf. Syst.* 41, 3, Article 75 (feb 2023), 27 pages. <https://doi.org/10.1145/3569930>
- [18] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) (WSDM '17). Association for Computing Machinery, New York, NY, USA, 781–789. <https://doi.org/10.1145/3018661.3018699>
- [19] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206. <https://doi.org/10.1109/ICDM.2018.00035>
- [20] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) (KDD '20). Association for Computing Machinery, New York, NY, USA, 1748–1757. <https://doi.org/10.1145/3394486.3403226>
- [21] Sen Li, Fuyu Lv, Taiwei Jin, Gulil 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* (Virtual Event, Singapore) (KDD '21). Association for Computing Machinery, New York, NY, USA, 3181–3189. <https://doi.org/10.1145/3447548.3467101>
- [22] Yiding Liu, Guan Huang, Jiaxiang Liu, Weixue Lu, Suqi Cheng, Yukun Li, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. *CoRR abs/2106.03373* (2021). arXiv:2106.03373 <https://arxiv.org/abs/2106.03373>
- [23] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisjuk. 2021. Que2Search: Fast and Accurate Query and Document Understanding for Search at Facebook. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) (KDD '21). Association for Computing Machinery, New York, NY, USA, 3376–3384. <https://doi.org/10.1145/3447548.3467127>
- [24] Hanqing Lu, Youna Hu, Tong Zhao, Tony Wu, Yiwei Song, and Bing Yin. 2021. Graph-based Multilingual Product Retrieval in E-commerce Search. arXiv:2105.02978 [cs.CL]
- [25] Alessandro Magnani, Feng Liu, Suthee Chaidaroon, Sachin Yadav, Praveen Reddy Suram, Ajit Puthenpuhussery, Sijie Chen, Min Xie, Anirudh Kashi, Tony Lee, and Ciya Liao. 2022. Semantic Retrieval at Walmart. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) (KDD '22). Association for Computing Machinery, New York, NY, USA, 3495–3503. <https://doi.org/10.1145/3534678.3539164>
- [26] Alessandro Magnani, Feng Liu, Min Xie, and Somnath Banerjee. 2019. Neural Product Retrieval at Walmart.Com. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) (WWW '19). Association for Computing Machinery, New York, NY, USA, 367–372. <https://doi.org/10.1145/3308560.3316603>
- [27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR abs/1301.3781* (2013). <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>
- [28] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian (Allen) Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic Product Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 2876–2885. <https://doi.org/10.1145/3292500.3330759>
- [29] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles R. Rosenberg. 2022. PinnerFormer: Sequence Modeling for User Representation at Pinterest. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022).
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 8024–8035. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [31] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 2227–2237. <https://doi.org/10.18653/v1/n18-1202>
- [32] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: Unbiased Boosting with Categorical Features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 6639–6649.
- [33] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.
- [34] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. Context and Attribute-Aware Sequential Recommendation via Cross-Attention. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) (RecSys '22). Association for Computing Machinery, New York, NY, USA, 71–80. <https://doi.org/10.1145/3523227.3546777>
- [35] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. In *Proceedings of the 26th ACM SIGKDD International*

- Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) (KDD '20). Association for Computing Machinery, New York, NY, USA, 3505–3506. <https://doi.org/10.1145/3394486.3406703>
- [36] Kyuyong Shin, Hanock Kwak, Su Young Kim, Max Nihlen Ramstrom, Jisu Jeong, Jung-Woo Ha, and Kyung-Min Kim. 2022. Scaling Law for Recommendation Models: Towards General-purpose User Representations. arXiv:2111.11294 [cs.IR]
- [37] Kyuyong Shin, Hanock Kwak, Wonjae Kim, Jisu Jeong, Seungjae Jung, Kyung-Min Kim, Jung-Woo Ha, and Sang-Woo Lee. 2022. Pivotal Role of Language Modeling in Recommender Systems: Enriching Task-specific and Task-agnostic Representation Learning. arXiv:2212.03760 [cs.IR]
- [38] Uriel Singer, Haggai Roitman, Yotam Eshel, Alexander Nus, Ido Guy, Or Levi, Idan Hasson, and Eliyahu Kiperwasser. 2022. Sequential Modeling with Multiple Attributes for Watchlist Recommendation in E-Commerce. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) (WSDM '22). Association for Computing Machinery, New York, NY, USA, 937–946. <https://doi.org/10.1145/3488560.3498453>
- [39] Aixin Sun. 2023. Take a Fresh Look at Recommender Systems from an Evaluation Standpoint. arXiv:2210.04149 [cs.IR]
- [40] 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* (Beijing, China) (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1441–1450. <https://doi.org/10.1145/3357384.3357895>
- [41] 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* (Marina Del Rey, CA, USA) (WSDM '18). Association for Computing Machinery, New York, NY, USA, 565–573. <https://doi.org/10.1145/3159652.3159656>
- [42] Yina Tang, Fedor Borisov, Siddharth Malreddy, Yixuan Li, Yiqun Liu, and Sergey Kirshner. 2019. MSURU: Large Scale E-Commerce Image Classification with Weakly Supervised Search Data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 2518–2526. <https://doi.org/10.1145/3292500.3330696>
- [43] 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*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [44] Binbin Wang, Mingming Li, Zhixiong Zeng, Jingwei Zhuo, Songlin Wang, Sulong Xu, Bo Long, and Weipeng Yan. 2023. Learning Multi-Stage Multi-Grained Semantic Embeddings for E-Commerce Search. *CoRR* abs/2303.11009 (2023). <https://doi.org/10.48550/arXiv.2303.11009> arXiv:2303.11009
- [45] Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. 2021. Personalized Embedding-based e-Commerce Recommendations at eBay. arXiv:2102.06156 [cs.IR]
- [46] Xuyang Wu, Alessandro Magnani, Suthee Chaidaroon, Ajit Puthenpuussery, Ciya Liao, and Yi Fang. 2022. A Multi-task Learning Framework for Product Ranking with BERT. In *Proceedings of the ACM Web Conference 2022*. ACM. <https://doi.org/10.1145/3485447.3511977>
- [47] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Max Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR* abs/1609.08144 (2016). <http://dblp.uni-trier.de/db/journals/corr/corr1609.html#WuSCLNMKCGMKSJL16>
- [48] Xue Xia, Neng Gu, Dhruvil Deven, and Andrew Zhai. 2022. *How Pinterest Leverages Realtime User Actions in Recommendation to Boost Homefeed Engagement Volume*. Retrieved March 2, 2023 from <https://medium.com/pinterest-engineering/how-pinterest-leverages-realtime-user-actions-in-recommendation-to-boost-homefeed-engagement-volume-165ae2e8cde8>
- [49] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [50] 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 Proceedings of the Web Conference 2020* (Taipei, Taiwan) (WWW '20). Association for Computing Machinery, New York, NY, USA, 441–447. <https://doi.org/10.1145/3366424.3386195>
- [51] 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* (Copenhagen, Denmark) (RecSys '19). Association for Computing Machinery, New York, NY, USA, 269–277. <https://doi.org/10.1145/3298689.3346996>
- [52] Han Zhang, Songlin Wang, Kang Zhang, Zhiling Tang, Yunjiang Jiang, Yun Xiao, Weipeng Yan, and Wen-Yun Yang. 2020. Towards Personalized and Semantic Retrieval: An End-to-End Solution for E-Commerce Search via Embedding Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) (SIGIR '20). Association for Computing Machinery, New York, NY, USA, 2407–2416. <https://doi.org/10.1145/3397271.3401446>
- [53] Jianjin Zhang, Zheng Liu, Weihao Han, Shitao Xiao, Ruicheng Zheng, Yingxia Shao, Hao Sun, Hanqing Zhu, Premkumar Srinivasan, Weiwei Deng, Qi Zhang, and Xing Xie. 2022. Uni-Retriever: Towards Learning the Unified Embedding Based Retriever in Bing Sponsored Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) (KDD '22). Association for Computing Machinery, New York, NY, USA, 4493–4501. <https://doi.org/10.1145/3534678.3539212>
- [54] Yunan Zhang, Le Yan, Zhen Qin, Honglei Zhuang, Jiaming Shen, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2023. Towards Disentangling Relevance and Bias in Unbiased Learning to Rank. arXiv:2212.13937 [cs.IR]
- [55] 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* (Copenhagen, Denmark) (RecSys '19). Association for Computing Machinery, New York, NY, USA, 43–51. <https://doi.org/10.1145/3298689.3346997>
- [56] Liying Zheng, Yingji Pan, and Yuri M. Brovman. 2022. Page-Wise Personalized Recommendations in an Industrial e-Commerce Setting. In *Proceedings of the 5th Workshop on Online Recommender Systems and User Modeling co-located with the 16th ACM Conference on Recommender Systems, ORSUM@RecSys 2022, Seattle, WA, USA, September 23rd, 2022 (CEUR Workshop Proceedings, Vol. 3303)*, João Vinagre, Marie Al-Ghossein, Alípio Mário Jorge, Albert Bifet, and Ladislav Peska (Eds.). CEUR-WS.org. <https://ceur-ws.org/Vol-3303/paper2.pdf>
- [57] Xiaoyang Zheng, Fuyu Lv, Zilong Wang, Qingwen Liu, and Xiaoyi Zeng. 2023. Delving into E-Commerce Product Retrieval with Vision-Language Pre-training. *CoRR* abs/2304.04377 (2023). <https://doi.org/10.48550/arXiv.2304.04377> arXiv:2304.04377
- [58] Xiaoyang Zheng, Zilong Wang, Sen Li, Ke Xu, Tao Zhuang, Qingwen Liu, and Xiaoyi Zeng. 2023. MAKE: Product Retrieval with Vision-Language Pre-training in Taobao Search. *CoRR* abs/2301.12646 (2023). <https://doi.org/10.48550/arXiv.2301.12646> arXiv:2301.12646
- [59] Yukun Zheng, Jiang Bian, Guang-Xun Meng, Chao Zhang, Honggang Wang, Zhixuan Zhang, Sen-Yuan Li, Tao Zhuang, Qingwen Liu, and Xiaoyi Zeng. 2022. Multi-Objective Personalized Product Retrieval in Taobao Search. *ArXiv* abs/2210.04170 (2022).
- [60] 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 Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (Honolulu, Hawaii, USA) (AAAI'19/IAAI'19/EAAI'19). AAAI Press, Article 729, 8 pages. <https://doi.org/10.1609/aaai.v33i01.33015941>
- [61] 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 and Data Mining* (London, United Kingdom) (KDD '18). Association for Computing Machinery, New York, NY, USA, 1059–1068. <https://doi.org/10.1145/3219819.3219823>