Self-Supervised Learning on Users' Spontaneous Behaviors for Multi-Scenario Ranking in E-commerce

Yulong Gu, Wentian Bao, Dan Ou, Xiang Li, Baoliang Cui, Biyu Ma,
Haikuan Huang, Qingwen Liu, Xiaoyi Zeng
Alibaba Group
guyulongcs@gmail.com
{wentian.bwt,oudan.od,leo.lx,moqing.cbl,biyu.mby,haikuan.hhk,xiangsheng.lqw,yuanhan}@alibaba-inc.com

ABSTRACT

Multi-scenario Learning to Rank is essential for Recommender Systems, Search Engines and Online Advertising in e-commerce portals where the ranking models are usually applied in many scenarios. However, existing works mainly focus on learning the ranking model for a single scenario, and pay less attention to learning ranking models for multiple scenarios. We identify two practical challenges in industrial multi-scenario ranking systems: (1) The Feedback Loop problem that the model is always trained on the items chosen by the ranker itself. (2) Insufficient training data for small and new scenarios. To address the above issues, we present **ZEUS**, a novel framework that learns a **Z**oo of ranking mod**E**ls for m**U**ltiple **S**cenarios based on pre-training on users' spontaneous behaviors (*e.g.,* queries which are directly searched in the search box and not recommended by the ranking system). ZEUS decomposes the training process into two stages: self-supervised learning based pre-training and fine-tuning. Firstly, ZEUS performs self-supervised learning on users' spontaneous behaviors and generates a pre-trained model. Secondly, ZEUS fine-tunes the pre-trained model on users' implicit feedback in multiple scenarios. Extensive experiments on Alibaba's production dataset demonstrate the effectiveness of ZEUS, which significantly outperforms state-of-the-art methods. ZEUS averagely achieves 6.0%, 9.7%, 11.7% improvement in CTR, CVR and GMV respectively than state-of-the-art method.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**.

KEYWORDS

Learning to Rank; Multi-Scenario; Intent Recommendation; Recommender System; E-commerce

ACM Reference Format:
Yulong Gu, Wentian Bao, Dan Ou, Xiang Li, Baoliang Cui, Biyu Ma,, Haikuan Huang, Qingwen Liu, Xiaoyi Zeng. 2021. Self-Supervised Learning on Users' Spontaneous Behaviors for Multi-Scenario Ranking in E-commerce. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3481953

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3481953

1 INTRODUCTION

Recommender Systems, Search Engines and Online Advertising are playing significant roles in E-commerce companies (*e.g.,* Amazon, Alibaba, JD.com) [14–16, 25, 50, 53]. For example, in Alibaba, one of the leading E-commerce companies in the world, the search and recommender systems serve more than 0.8 billion users, and contribute over a trillion dollars of Gross Merchandise Volume (GMV) (*i.e.,* the total sales value for merchandise sold) in 2020. Industrial search and recommender systems usually consist of two stages: The first stage is retrieval (or candidate generation) [3, 7, 26, 30, 41], which selects hundreds or thousands of items as candidates from millions or billions of items. The second stage is ranking [7, 50], which predicts the ranking scores of the selected candidates and returns top-ranked items. In this paper, we focus on the Learning to Rank (LTR) problem in the ranking stage. LTR aims to learn a ranking model which scores the candidate items given the context information. Depending on the type of the item and context information, a typical ranking task in e-commerce could be: (1) Intent Recommendation, when the item is a query; (2) Product Recommendation, when the item is a product; (3) Product Search, when the item is a product and the context information is a query.

Existing industrial LTR approaches, such as traditional machine learning models (*e.g.,* LR [19, 31], GBDT [18, 46]) and deep neural networks based approaches (DNN [7], Wide & Deep [6], DIN [50], DIEN [49], DMT [15]), usually assume that the data are in the same distribution and focus on learning the ranking model based on users' implicit feedback data in a single scenario. However, in real-world platforms such as Alibaba, the ranking models usually serve in different scenarios. For example, as shown in Figure 1, Intent Recommendation is widely used in different scenarios (such as Homepage, Search Discovery, landing page from Affiliated APPs, Taobao Special Price) in Alibaba's Taobao [1] app, which is one of the largest online shopping application in the world. The training data in different scenarios may come from different distributions because the users and items may differ greatly. Training scenario-specific ranking model only based on the data in each scenario is not suitable for small or new scenarios where feedback data is

[1] www.taobao.com

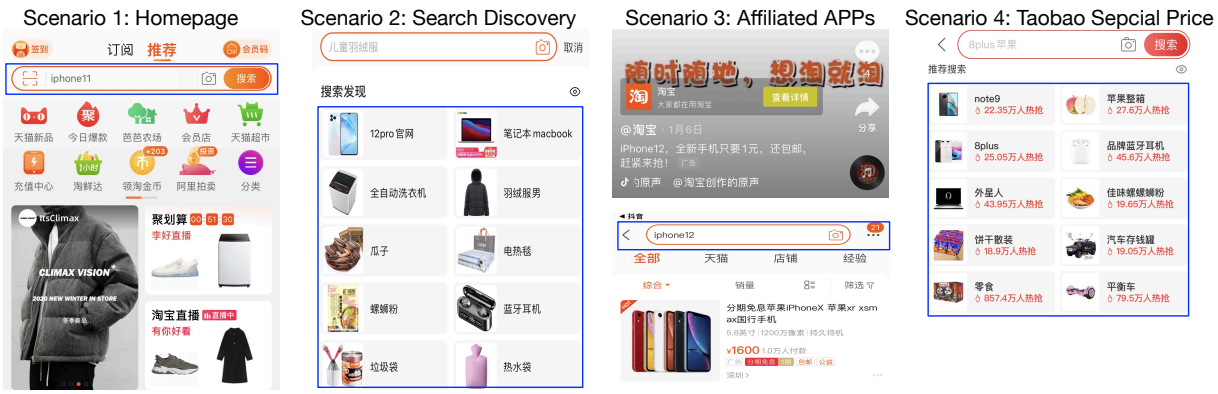

Figure 1: Multi-Scenario Intent Recommender System in Alibaba. The queries in in the blue boxes are recommended intents.

very limited, while simply mixing all the data and training a shared ranking model cannot capture the characteristics of each scenario.

Consequently, Multi-scenario Learning to Rank, which aims to capture both the commonalities and scenario-specific characteristics of multiple scenarios, is essential in real-world web services. Designing a Multi-Scenario Ranking system faces two challenges:

- **The Feedback Loop Problem.** Existing LTR approaches usually train the ranking model based on users' implicit feedback on the items, which are selected by the old ranking model. The biased exposure of items will lead to the Feedback Loop Problem [39] in industrial ranking systems, where popular items become even more popular and long-tail items become even less popular. Such phenomenon will make ranking models generate increasingly biased results over time. Therefore how to address the feedback loop problem is a critical issue in industrial ranking systems.
- **Insufficient training data for small and new scenarios.** State-of-the-art LTR approaches are based on DNNs with millions or even billions of parameters [6, 7, 15, 49, 50]. In real-world platforms, many small and new scenarios have rather limited data. Training giant neural networks solely based the scenario's own data may lead to overfitting and inferior performance.

Some pioneering studies [5, 22, 37] model the Multi-Scenario LTR as a multi-task problem and focus on learning the relationships of the training data in multiple scenarios. However, such methods only utilize users' implicit feedback (*e.g.,* clicked queries from the Intent Recommender System) and neglect users' spontaneous behaviors (*e.g.,* directly searched queries in the search box) in the application, therefore they may suffer from severe feedback loop problem. For example, for the e-commerce search engine in Alibaba, there are two types of query sources: users' spontaneously searched queries (*i.e.,* the users directly search the queries in the search box, *e.g.,* q_1, q_3, q_t in Figure 2) and users' clicked queries that are recommended by the Multi-scenario Intent Recommender System (*e.g.,* q_2, q_4 in Figure 2). For the e-commerce search engine in Taobao, users' spontaneously searched queries and clicked queries in the Multi-scenario Intent Recommender System contribute about 73% and 27% respectively. In this paper, we find that modeling users' large-scale spontaneous behaviors is extremely important for multi-scenario ranking.

To address these problems, we propose **ZEUS**, a novel framework that can jointly learn a Zoo of ranking models for multiple scenarios based on pre-training on users' spontaneous behaviors. As illustrated in Figure 2, the training procedure of ZEUS consists of two stages: self-supervised learning based pre-training and fine-tuning. In each of these stages, we use the Sequential Interest Model, which models users' interest based on the input features (*e.g.,* sequential behaviors), as the ranking model. The key idea of ZEUS is exploiting pre-training to improve the learning of the Sequential Interest Model. In the first stage, ZEUS performs self-supervised learning on users' spontaneous behaviors and generates a task-agnostic pre-trained model. Specifically, We define a pretext task called Next-Query Prediction that aims to predict the user's next spontaneous searched query, which is the ideal query that the Intent Recommender System should recommend. This pre-training stage can improve the understanding of users' search behaviors. In the fine-tuning stage, ZEUS firstly captures the commonalities of different scenarios by fine-tuning the pre-trained model on users' feedback in multiple scenarios simultaneously, and then models the scenario-specific characteristics of each scenario by further fine-tuning using the implicit feedback data in each scenario. Extensive experiments on the industrial Alibaba's Multi-Scenario Intent Recommendation dataset demonstrate the effectiveness of ZEUS.

Our major contributions are as follows:

- We highlight the significance of pre-training on users' spontaneous behaviors, which can alleviate the long-standing Feedback Loop problem in learning ranking models, solve the insufficient training data problem in small and new scenarios, and improve the ranking performance in all scenarios.
- We present a novel framework ZEUS, which learns a Zoo of ranking modEls for mUltiple Scenarios based on pre-training.
- We conduct extensive experiments and demonstrate that ZEUS significantly outperforms state-of-the-art baselines for ranking in multiple scenarios (*e.g.,* large, small and new scenarios). ZEUS averagely increases the CTR, CVR and GMV by 6.0%, 9.7%, 11.7% respectively, and has been deployed in the commercial Multi-Scenario Intent Recommendation System in Taobao, contributing billions of dollars in revenue for Alibaba each year.

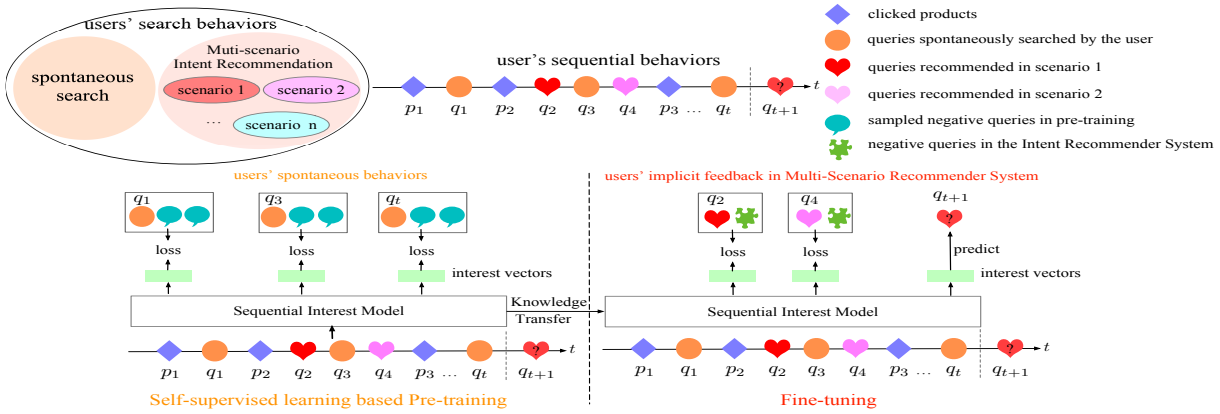

Figure 2: The training procedure of ZEUS. It utilizes Self-supervised Learning based pre-training and fine-tuning for ranking.

2 RELATED WORK

2.1 Learning to Rank

Learning to Rank, which aims to learn a ranking model, is a significant task in industrial applications, such as recommender systems, search engines, online advertising, and so on. Traditional machine learning models, such as Logistic Regression (LR) [19, 31] and Gradient Boosting Decision Tree (GBDT) [12, 18, 46], are popular and successful methods for learning ranking models in industrial systems. In recent years , Deep Neural Networks based methods, such as DNN [7] and Wide & Deep [6], have achieved appealing performance in ranking. These methods follow the Embedding & MLP paradigm, where large-scale sparse features are firstly embedded into low dimensional vectors, and then concatenated together to fed into the multilayer perceptron (MLP) to learn the nonlinear relations among features. State-of-the-art LTR methods find the effectiveness of extracting users' interest from their historical behaviors for ranking. To be specific, DIN [50] uses attention mechanism to learn the representation of users' interest from users' historical behaviors with respect to a candidate item. DIEN [49] and HUP [14] uses recurrent neural networks to capture the evolution of users' interests. DMT [15] exploits multiple transformers to model users' diverse behavior sequences and achieves state-of-the-art performance in recommendation [1, 15]. These methods focus on learning the ranking models based on users' implicit feedback in a single scenario, pay few attention to learning ranking models for multiple scenarios and neglect users' spontaneous behaviors.

2.2 Multi-Scenario Learning to Rank

Multi-Scenario Learning to Rank, which aims to learn ranking models for multiple scenarios, is essential and significant for industrial applications, where ranking services are usually applied in different scenarios. There are some pioneering work [5, 22, 37] that starts to investigate this problem. HMoE [22] formulates this problem as a multi-task learning problem, and uses Multi-task Mixture-of-Experts [27, 47] to implicitly identify distinctions and commonalities between tasks, and improves the performance with a stacked model learning task relationships in the label space explicitly. SAML [5] focuses on modeling the difference and similarities between multiple scenarios. Although these methods are beneficial to improve the performance of ranking in multiple scenarios, they only exploit users' implicit feedback data in the recommender systems and neglect users' spontaneous behaviors. These methods are orthogonal to our approach, which focuses on pre-training on users' spontaneous behaviors. We leave the combination of these approaches with our method as future work.

Cross-domain Recommendation [10, 20, 23, 33] aims to improve the ranking performance of a type of items in the target domain by transferring knowledge from other types of items in source domains. For example, Ouyang et al. [33] proposed the Mixed Interest Network, which improves the CTR prediction of a target domain (i.e. ads) leveraging auxiliary data from a source domain (i.e. news). It is significant different with the Multi-Scenario Learning to Rank problem, which aims to improve the ranking performance of the same type of items in different scenarios.

2.3 Self-supervised Learning

Self-supervised Learning (SSL) [2, 34], which aims to enhance the represent learning based on unlabeled data, has been widely used in the areas of Compute Vision (CV) [2, 4, 34] and Natural Language Processing (NLP) [8, 17] area. The basic idea is to define a supervised pretext task based on the unlabeled dataset. After the pretext task training finished, the learned parameters serve as a pre-trained model and are transferred to downstream tasks by fine-tuning. For example, in CV, Pathak et al. [34] proposed to generate the contents of an arbitrary image region conditioned on its surroundings. SimCLR [2] proposed a simple framework for contrastive learning of visual representations. SimSiam [4] proposed a simple Siamese network, which can learn meaningful representations without using negative sample pairs, large batches and momentum encoders. In NLP, BERT [8] pretrains deep bidirectional representations from unlabeled text. The pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of downstream tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

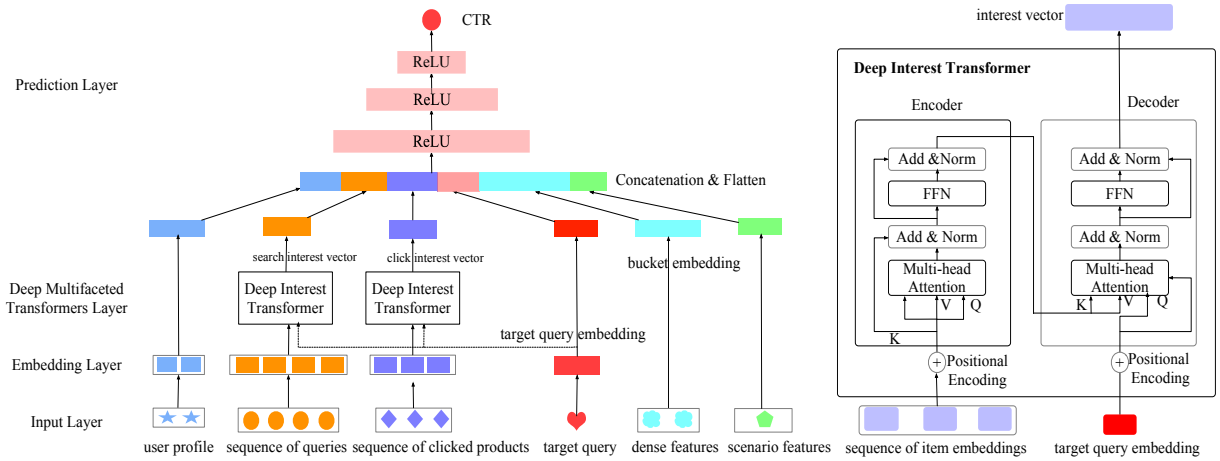

Figure 3: The Sequential Interest Model DMT in ZEUS. It utilizes Deep Multifaceted Transformers (bottom), which is consisted of multiple Deep Interest Transformers (right), to extract users' multifaceted interests from their diverse behavior sequences.

Recently, there are some pioneering work [28, 38, 42, 43, 45, 48, 51] that attempt to apply self-supervised learning in Recommender Systems. For the matching task, some researchers[28, 45] exploit self-supervised learning to learn better representations for long-tail items and achieve better performance in candidate generation. For the ranking task, S3-Rec [51] devises four auxiliary self-supervised objectives on the training data to learn the correlations among attribute, item, subsequence, and sequence by utilizing the mutual information maximization (MIM) principle. These self-supervised learning based ranking approaches focus on utilizing users' implicit feedback data in the recommender system and neglect users' spontaneous behaviors. In this paper, we proposed the idea of performing self-supervised learning on users' spontaneous behaviors to improve the performance and break the Feedback Loop problem in ranking. This is significantly different with these approaches.

2.4 Intent Recommendation

Intent Recommendation (or Query Recommendation) [11, 44], which aims to predict users' search intent and recommend potentially interested queries to users, is playing significant roles in the growth of search in e-commerce. MEIRec [11] models the relationships between users, products and queries as a heterogeneous graph and uses a Metapath-guided Heterogeneous Graph Neural Network for intent recommendation. FINN [44] uses the feedback interactive neural network to model both the positive feedback and negative feedback information simultaneously and achieves state-of-the-art performance. The Intent Recommendation problem is different with both Query Suggestion [24] and Related Query Recommendation [21, 36, 52], where a partial or complete input query is needed. Query Suggestion [24] aims to recommend some queries that starts with the partial query (*i.e.,* query prefix) from the user. Related Query Recommendation [21, 36] aims to recommend some related queries that the user would be interested in based on the user's current input query.

3 PROBLEM FORMULATION

The Multi-Scenario Learning to Rank problem. Given a set of scenarios $\mathcal{R} = \{\mathcal{R}_k\}_{k=1}^{|\mathcal{R}|}$, they share a common features space $\mathcal{F}$ and label space $\mathcal{Y}$. For scenario $\mathcal{R}_k$, the labeled training data are: $D_{\mathcal{R}_k} = \{(f_i, y_i)\}, i = 1, 2, ..., |D_{\mathcal{R}_k}|$, which are drawn from a domain-specific distribution $P_{\mathcal{R}_k}$ over $\mathcal{F} \times \mathcal{Y}$. For different scenarios, the distribution $P_{\mathcal{R}_k}$ are different. The Multi-Scenario Learning to Rank problem aims to learn a zoo of ranking models $\mathcal{M} = \{\mathcal{M}_k\}_{k=1}^{|\mathcal{R}|}$, where each scenario-specific ranking model $\mathcal{M}_k$ is used for ranking in scenario $\mathcal{R}_k$.

4 METHOD

In this section, we introduce the details of our framework ZEUS, which aims to build a Zoo of ranking models for multiple scenarios based on pre-training. There are two stages in ZEUS: self-supervised learning based pre-training and fine-tuning. These two stages both use the Sequential Interest Model to model users' interests. In the pre-training stage, ZEUS performs self-supervised learning on users' spontaneous behaviors. In the fine-tuning stage, ZEUS fine-tunes the pre-trained model based on users' implicit feedback in the Multi-Scenario Intent Recommender System.

4.1 Sequential Interest Model

The Sequential Interest Model, which learns the representation of users' interest based on their historical behaviors, is used as the backbone ranking model [49, 50]. In this paper, we use Deep Multifaceted Transformers (DMT) [15] (shown in Figure 3) as the Sequential Interest Model. DMT is the state-of-the-art ranking model that serves the main traffic in recommender systems and search engines in large-scale e-commerce sites like Taobao and JD.com [1, 15].

4.1.1 Input and Embedding Layer. The inputs can be divided into two parts: categorical features and dense features.

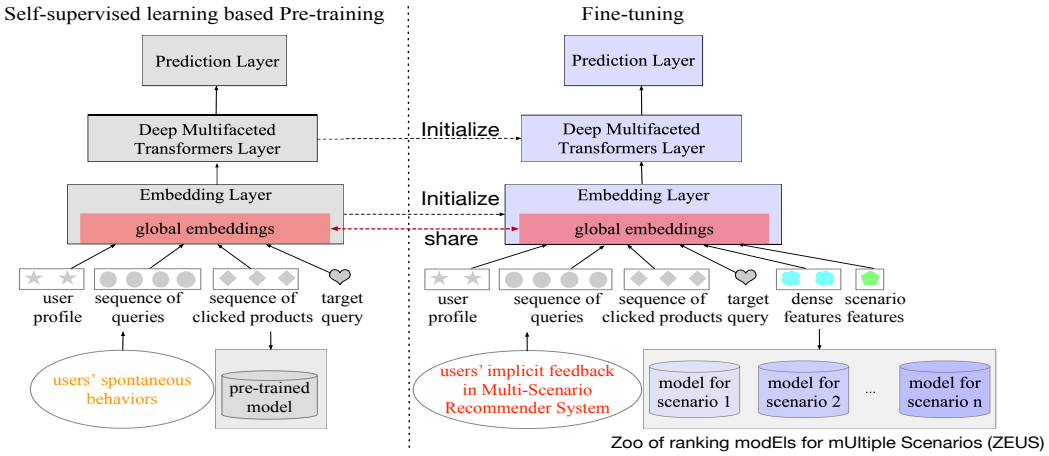

Figure 4: The Architecture of ZEUS.

Categorical features. The categorical features are user profile, users' diverse behaviors, item profile and scenario-specific features.

- **User Profile**. User Profile contains user's characteristics, such as age, gender, purchase power and so on.
- **User's Behaviors Sequences**. For the behaviors sequences, we empirically find that the most useful sequences for Intent Recommendation are the query sequence S_q (*i.e.,* the sequence of users' historical queries) and click sequence S_c (*i.e.,* the sequence of users' clicked products). A user's behavior sequence is represented by a variable-length sequence $S = \langle x_1, x_2, \ldots, x_T \rangle$, where T is the length of the sequence.
- **Item Profile**. In the Intent Recommendation problem, there are two types of items: queries and products . For each query, we use its query id, characters, word segments, predicted product category ids and statistical features (*e.g.,* click-through rate) to represent it. For each product, we use its product id, category id, brand id and shop id to represent it.
- **Scenario features**. The scenario features include the scenario information (*e.g.,* scenario names).

Dense features. The dense features are numerical features which are used by the last generation Intent Recommender System (*i.e.,* FINN [44]) in Alibaba.

As previous work did [15], we use embedding layers to transform the high dimensional sparse ids and dense features into low dimensional dense vectors.

4.1.2 Deep Multifaceted Transformers Layer. To capture the user's interest, like the state-of-the-art ranking model DMT [15] did, we use two separate Transformers [13, 40] to model the user's query sequence and click sequence, and learn the user's interest vectors in different perspective respectively. The basic idea is that users' multiple types of behavior sequences on queries and products are significantly different and they have different timescales. For each behavior sequence, we exploit Transformer (right side in Figure 3) to model user's real-time interest and represent it as an interest vector. In the Transformer, the encoder models the relationships among items in the sequence, and the decoder learns user's interest vector corresponding to the target item. The encoder applies self-attention on the the embeddings of the behavior sequence and allows each item in the sequence to attend over all items in the input sequence. As a user may have diverse interest [50], the decoder uses the target item as query and the output of the encoder as both keys and values, exploits target attention to learn the attention score between the target item and each item in the historical sequence, and learns a unique user interest vector for each target item.

4.1.3 Prediction Layer. For each input sample, the Prediction Layer uses Multi-layer Preceptrons (MLP) to predict the Click-Through Rate (CTR) of the item:

$$\hat{y} = SIM(f) = MLP(DMT(f)) \tag{1}$$

where SIM denotes the Sequential Interest Model, f is the input features, $DMT(f)$ is the output from the Deep Multifaceted Transformers Layer, and $\hat{y}$ is the predicted CTR.

4.2 Self-supervised learning based Pre-training

This stage performs self-supervised learning on the large corpus of user's spontaneous behaviors to improve the representation learning on users' behaviors, and learns a task-agnostic pre-trained Sequential Interest Model.

Self-supervised learning [2, 8] has achieved state-of-the-art performance in Natural Language Processing and Computer Vision. The key idea in Self-supervised learning is the designing of Pretext (*i.e.,* pre-training) task, which aims to generate supervised training data from unlabeled data by predicting some part of the input from the remaining part. For the Intent Recommendation problem, the main input of the Sequential Interest Model is users' sequential behaviors. So a natural idea is learning to predict a user's some behaviors from the remaining behaviors in the sequence. Inspired by BERT [8] and GPT [35], which are the state-of-the-art self-supervised learning methods in the Natural Language Processing area, we have designed one Pretext task called Next-Query Prediction (NQP) in ZEUS.

Pretext Task: Next-Query Prediction (NQP). The NQP task aims to predict the next query that a user will spontaneously search in the search box based on her historical behaviors. For a behavior sequence $S = \langle x_1, x_2, \ldots, x_T \rangle$, where T is the length of the sequence and x_i indicates a behavior on the item x_i at the time stamp i, the goal of the NQP task is to predict that the user will spontaneously search query x_{t+1} at the next step based on the user's current context state $c_t = \{x_1, x_2, ..., x_t\}$. For example, as illustrated in the left part of Figure 2, the NQP task aims to predict that the user will search queries q_1 , q_3, q_t after she performs behaviors on p_1, q_2 and p_3 respectively.

Loss Function. BERT [8] and GPT [35] exploit softmax function to predict the probability distribution of next item (*i.e.,* token), where the vocabulary sizes of items are only about 30,000 [8] and 40,000 [35] respectively. However, the softmax function is not suitable for industrial ranking system, where the vocabulary size of items (*e.g.,* queries, products) can be several million or billion. To solve this high-dimensional sequential modeling problem, as Contrastive Predicting Coding (CPC) [32] did, we can encode the current and future information as distributed vector representations and use InfoNCE loss to maximize the mutual information between the current context c_t and future target x_{t+1} defined as

$$I(x_{t+1}; c_t) = \sum_{x_{t+1}, c_t} p(x_{t+1}, c_t) log \frac{p(x_{t+1}|c_t)}{p(x_{t+1})} \tag{2}$$

where p is the probability distribution function. Given a set $X_N = \{x_1, x_2, ..., x_N\}$ of N random samples containing one positive sample from $p(x_{t+1}|c_t)$ and $N-1$ negative samples from the distribution $p(x_{t+1})$, the self-supervised InfoNCE loss is defined as:

$$L_{self-supervised} = -\mathbb{E}_{X_N}[log(\frac{f(x_{t+1}, c_t)}{\sum_{x_j \in X_N} f(x_j, c_t)})] \tag{3}$$

where f is the density ratio. In ZEUS, we use the Sequential Interest Model (SIM) to calculate $f(x_j, c_t) = SIM((x_j, c_t))$ where x_j is a candidate item, and $c_t = \{x_1, x_2, ..., x_t\}$ is the user's recent behaviors sequence. The authors in [51] demonstrated the InfoNCE can be approximated by cross-entropy loss. We empirically find the effectiveness of using this approximate method.

Negative examples sampling. We treat users' search queries as positive examples and sample $N-1$ negative queries for each positive query. There can be hundreds of millions of queries in industrial system and the selection of negative examples is extremely important for training models. We find it effective to treat suggested queries that are exposed but not clicked as negative examples.

Discussion. Compared with BERT and GPT, our method is different as follows: (1) ZEUS is light-weighted and time-efficient. The time cost of ZEUS is only about 25 ms in online serving. BERT and GPT use more than 12 layers of Transformers, which will lead to heavy cost of time. They cannot be directly used for ranking in industrial systems, which have strict limitation of time cost (*i.e.,* lower than 100 ms). (2) BERT and GPT need the input is in the format of a sequence, which is not suitable for the industrial ranking features, where other non-sequential types of features (*e.g.,* user profiles, dense features) also exist. (3) BERT only uses the encoder in Transformer and GPT only uses the decoder in Transformer. Our method uses both encoder and decoder, which is stronger for next item prediction [15]. What's more, the bidirectional encoder in BERT is not suitable to model the chronological characteristics of users' behaviors in real-world applications. We have also tried the state-of-the-art self-supervised learning method (*i.e.,* Simsiam [4]) in Computer Vision, but find that it does not bring additional gain.

4.3 Fine-tuning.

In the fine-tuning stage, we fine-tune the pre-trained model and generate the zoo of ranking models for multiple scenarios. To be specific, firstly, we fine-tune the pre-trained Sequential Interest Model based on the implicit feedback data in all the scenarios and obtain a task-adaptive ranking model, which can be directly used for ranking in new ranking scenarios where there is no training data. Secondly, for each scenario, we fine-tune the task-adaptive ranking model using the implicit feedback data in this scenario and obtain the scenario-specific ranking model. For example, we obtain the ranking model for scenario 1 by fine-tuning the task-adaptive ranking model using implicit feedback data in scenario 1.

We empirically find that the performance of ZEUS is the best when the sequential interest models in the pre-training and fine-tuning stages share the same global embeddings, which contain embedding vectors for high dimensional sparse ids with large vocabulary size (*i.e.,* query ids and products ids). Consequently, in the fine-tuning stage, we fix the global embedding and only fine-tune other parameters (*i.e.,* embedding vectors of other sparse ids, parameters in the Deep Multifaceted Transformer Layer and the Prediction Layer). The multiple ranking models for different scenarios use the shared global embedding parameters to represent the commonalities of multiple scenarios, and other scenario-specific parameters to capture the characteristics of each scenario.

In the fine-tuning stage, we use the supervised cross entropy loss function $L_{supervised}$ for model training:

$$L_{supervised} = -\frac{1}{Z} \sum_{i=1}^{Z} (y_i log \hat{y}_i + (1 - y_i) log(1 - \hat{y}_i)) \tag{4}$$

where Z is the size of training set, $y_i \in \{0, 1\}$ is the ground truth label, and $\hat{y}_i$ is the predicted CTR.

4.4 Model Training and Prediction.

In the training stage, the loss function L is defined as:

$$L = L_{self-supervised} + L_{supervised} \tag{5}$$

where $L_{self-supervised}$ and $L_{supervised}$ denote the losses in the pre-training and fine-tuning stages respectively.

In the prediction stage in online serving system, we use the predicted CTR $\hat{y}_i$ (in Equation 4) as the ranking score of the input.

5 EXPERIMENTS

5.1 Dataset

We conduct our research on Alibaba's Multi-Scenario Intent Recommendation dataset (Ali-MSIR), which is collected from the implicit feedback logs in Taobao Multi-Scenario Intent Recommender System from Nov 1, 2020 to Dec 31, 2020. The data from Nov 1, 2020 to Dec 30, 2020 is used as the training set, and the data from Dec 31,

Table 1: Statistics of the Dataset

Data Source	Samples	Positive Samples
Users' Spontaneous Behaviors	15.47 billion	1.55 billion
Scenario 1: Homepage	4.96 billion	0.76 billion
Scenario 2: Search Discovery	0.47 billion	0.17 billion
Scenario 3: Affiliated APPs	2.21 billion	0.29 billion

Table 2: Performance of different methods for Intent Recommendation. "$*$" indicates the statistically significant improvements (i.e., p-value < 0.01) over the best baseline.

Model	AUC		
	Scenario 1	Scenario 2	Scenario 3
GBDT	0.754	0.805	0.767
DNN	0.807	0.815	0.768
FINN(Base)	0.808	0. 818	0.769
DMT	0.810	0.820	0.771
ZEUS	**0.826***	**0.854***	**0.785***

2020 is used as the testing set. We also sampled Users' Spontaneous Search Behaviors in Taobao Search from Dec 1, 2020 to Dec 30, 2020. For each positive search query of a user, we sample 9 queries which are exposed to her but not clicked as negative examples. The statistics of the dataset is shown in Table 1. We evaluate the models on one large scenario ("Homepage"), two small scenarios ("Search Discovery" and "Affiliated APPs") and one new scenario ("Taobao Special Price") where there is no training data.

5.2 Baselines

- **GBDT** [12]. GBDT, which is a widely used model for industrial recommender systems, was used for Intent Recommendation in Taobao before 2019.
- **FINN** [44]. FINN, which uses the feedback interactive neural network to model both the positive and negative feedback simultaneously, achieves state-of-the-art performance for Intent Recommendation. It was used as the last generation Intent Recommendation method at Taobao from 2019 to 2020.
- **DNN** [7]. DNN uses pooling operation to aggregate users' sequential behaviors and exploits Deep Neural Networks for ranking.
- **DMT** [15]. DMT exploits multiple transformers to model users' diverse types of behaviors, and achieves state-of-the-art performance for ranking in Recommender Systems. DMT is a special case of ZEUS where the pre-training stage is not used.

5.3 Evaluation Metrics

To evaluate the effectiveness of the methods, for offline A/B Testing, we use the widely used metric AUC [50]. For online A/B Testing, we use three core metrics in e-commerce: CTR, CVR and GMV [50].

6 EXPERIMENTAL RESULTS

We aim to answer the follow questions:

Table 3: Influence of different types of behavior sequences.

Model	behaviors	AUC		
		Scenario 1	Scenario 2	Scenario 3
ZEUS	no	0.780	0.812	0.768
ZEUS	query	0.822	0.825	0.781
ZEUS	click	0.818	0.845	0.782
ZEUS	query+click	**0.826***	**0.854***	**0.785***

- **RQ1**: How does ZEUS perform compared with state-of-the-art methods for multi-scenario ranking?
- **RQ2**: How do different components affect the performance of ZEUS?
- **RQ3**: How does ZEUS perform in real-world multi-scenario recommender systems in e-commerce?
- **RQ4**: Can ZEUS break the Feedback Loop problem?
- **RQ5**: Can ZEUS provide meaningful interpretation of the recommendation results?

6.1 Comparison with Baselines (RQ1)

Table 2 shows the experimental results of different methods for the Multi-Scenario Intent Recommendation task. All experiments are repeated 5 times and the averaged results are reported. From this table, we can find that: (1) DNN performs better than GBDT by modeling the sequential behaviors. FINN can achieve better performance than DNN by using the attention mechanism. DMT improves the performance further by modeling the sequential behaviors using multiple transformers. (2) Our method ZEUS achieves superior performance compared with DMT by using pre-training. Compared to the state-of-the-art method DMT, ZEUS achieves 0.016, 0.034, 0.014 absolute AUC gains for one large scenario (*i.e.,* scenario 1) and two small scenarios (*i.e.,* scenario 2 and 3) respectively. These are significant improvements for industrial applications where 0.001 absolute AUC gain is remarkable [15, 29, 50].

6.2 Ablation Study (RQ2)

To investigate the effectiveness of components in ZEUS, we conduct multiple ablation studies.

6.2.1 Sequential Interest Model. Firstly, we investigate how different types of sequential behaviors in the Sequential Interest Model influence the performance of the ZEUS, and list the results in Table 3. From this table, we can find that: Both query sequence and click sequence are important for Intent Recommendation, and modeling them simultaneously achieves the best performance. We empirically find that further adding other types of behaviors (*e.g.,* add to cart, order [15]) doesn't bring additional gain. So we use the query and click sequences in ZEUS.

6.2.2 Self-supervised Learning based Pre-training Layer. Secondly, we investigate how the Self-supervised Learning Pre-training Layer influence the performance of ZEUS by comparing ZEUS with DMT, which is a special case of ZEUS where the pre-training stage is not used. From the fourth and fifth rows in Table 2, we can find that:

Table 4: Performance in online A/B Testing in Alibaba's Multi-Scenario Intent Recommender System. "*" indicates the statistically significant improvements (i.e., p-value < 0.01) over the baseline.

Model	Scenario 1			Scenario 2			Scenario 3			Scenario 4		
	CTR	CVR	GMV	CTR	CVR	GMV	CTR	CVR	GMV	CTR	CVR	GMV
GBDT	-1.9%	-5.0%	-7.3%	-0.4%	-2.0%	-4.3%	-3.4%	-1.1%	-2.8%	-6.7%	-11.5%	-15.3%
FINN(base)	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%	+0.0%
DMT	+3.6%	+1.6%	+1.8%	+3.7%	+2.6%	+1.9%	+1.4%	+1.1%	+2.4%	+1.4%	+2.5%	+3.1%
ZEUS	**+16.6%***	**+22.9%***	**+28.4%***	**+6.1%***	**+8.2%***	**+7.1%***	**+5.9%***	**+5.7%***	**+12.0%***	**+5.3%***	**+9.8%***	**+8.4%***

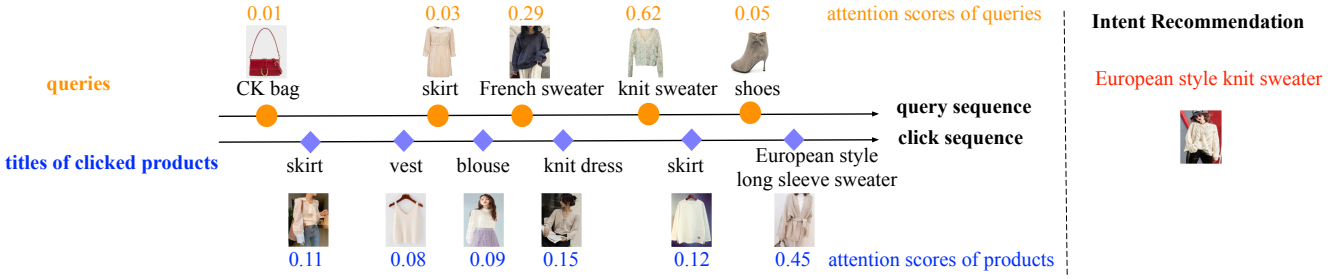

Figure 5: Case study of ZEUS.

the Self-supervised Learning based Pre-training stage can improve the ranking performance in all scenarios.

6.3 Online A/B Testing (RQ3)

To further evaluate the performance of ZEUS, we conduct online A/B testing for one month. The results are shown in Table 4. From this table, we can find that: (1) FINN averagely outperforms GBDT by 3.1%, 4.9% and 7.4% in CTR, CVR and GMV respectively in these scenarios. This demonstrates the effectiveness of modeling users' sequential behaviors. (2) DMT averagely outperforms FINN by 2.5%, 2.0% and 2.3% in CTR, CVR and GMV respectively in these scenarios. This illustrates that the self-attention and target-attention is essential for sequential modeling. (3) ZEUS averagely outperforms state-of-the-art method DMT by 6.0%, 9.7% and 11.7% in CTR, CVR and GMV respectively. This demonstrates the significance of the pre-training in ZEUS. It can solve the insufficient training data problem in small and new scenarios and improve the performance in all scenarios. (4) Compared with FINN, the last generation model in our Intent Recommender System, ZEUS averagely improves the online CTR, CVR and GMV by 8.5%, 11.7% and 14.0% respectively, which are the largest improvements in Alibaba's Intent Recommender System over past five years. In a word, ZEUS outperforms state-of-the-art methods for all scenarios, and it has been deployed online successfully and serves the main traffic in Alibaba's Multi-Scenario Intent Recommender System.

6.4 Breaking the Feedback Loop problem (RQ4)

To investigate whether our method can alleviate the Feedback Loop problem, we analysis the number of unique queries that are recommended to the users by the models in a month, and demonstrate the results in Table 5. From this table, we find that: ZEUS can averagely improves the number of unique queries by 20.2% leveraging the pre-training stage. This means that ZEUS can recommend more long-tail queries to the users and break the Feedback Loop problem.

Table 5: Performance in breaking Feedback Loop problem.

Pre-training	Number of Unique Queries		
	Scenario 1	Scenario 2	Scenario 3
no	11.03 million	11.57 million	2.28 million
has	13.74 million	14.06 million	2.61 million
Improve	+24.6%	+21.5%	+14.5%

6.5 Case Study (RQ5)

In this section, we investigate whether ZEUS can provide meaningful interpretation of the recommendation results using case study. From Figure 5, we find that: For the queries in the query sequence, the queries "French sweater" and "knit sweater" are more similar to the recommended query "European style knit sweater", and they have much higher attention scores than other queries. For the products in the click sequence, the products "knit dress" and "European style long sleeve sweater" have stronger relationship with the recommended query, and they have much higher attention scores than other products. By identifying that the user's main interest is "sweater" and she is interested in the "French", "European", "knit" styles, ZEUS recommends the query "European style knit sweater" to her. This demonstrates that ZEUS can accurately model users' interest from multiple types of behavior sequences, and it has good interpretation ability for the recommendation results.

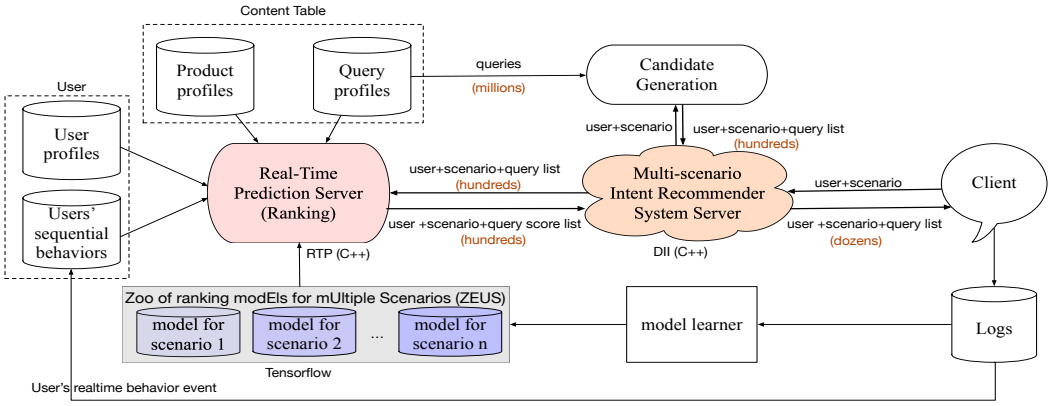

Figure 6: The System Architecture of ZEUS.

7 IMPLEMENTATION DETAILS

7.1 Implementation specifications

We have implemented our framework ZEUS with TensorFlow. For the Sequential Interest Model, the hidden size, number of heads, number of blocks in the Transformer are 128, 8 and 1. For each user, we select her recent 10 behavior for each type of behaviors. For the Prediction Layer, the layers of MLP are 512×256×64×1. The mini-batch size is 1024. We use Adagrad [9] as the optimizer and the learning rate starts at 0.01. The total number of parameters of ZEUS is about 6.43 billion. We use the Parameter Server framework called AOP in Alibaba to train the models on CPUs, where the servers have 50×30 CPU cores and the workers have 1000×60 CPU cores.

7.2 Sequential Interest Model Settings

7.2.1 Dense Features. The last generation Intent Recommender System in Alibaba exploits some dense features, which have been designed and improved for more than five years, to learn the ranking models such as GBDT [12] and FINN [44]. We empirically find that it will bring significant gains by integrating these dense features into our model. We use 49 dense features, which can mainly be divided into four types: item profile features (*e.g.,* number of clicks, CTR), use profile features (*e.g.,* purchase power), user-item matching features (*e.g.,* whether the item matches the user's gender or age) and user-item interaction features (*e.g.,* number of clicks on the category of the item within a time window).

7.2.2 Categorical features. We use the embedding layer to represent the inputs as low dimensional vectors. As previous work did [7, 15, 50], for each item, we use the embedding layer to transform the high dimensional sparse ids of the item's attributes into low dimensional dense representations and then concatenate these vectors into a single embedding vector to represent the item. The statistics of main embedding tables in ZEUS is shown in Table 6.

7.3 System Architecture

The system architecture of the Multi-Scenario Intent Recommender System is shown in Figure 6. When a user enters a scenario, the client will issue a request with the user and scenario information to the Multi-scenario Intent Recommender System Server, where the recommendation service is written with the C++ framework called DII in Alibaba. Firstly, the recommendation service will call the Candidate Generation module to generate hundreds of candidate queries. Secondly, the recommendation service will send the candidate queries to the Real-Time Prediction (RTP) Server to predict the ranking scores. The RTP service will use the scenario's corresponding ranking model in ZEUS to calculate the ranking scores. Thirdly, the recommendation service will select dozens of top ranked queries to the client. The QPS (Queries per second) of the Alibaba's Multiple-scenario Intent Recommender System is about 50,000. For the ranking models in ZEUS, the average RT (*i.e.,* Response Time, or time cost) is about 25 ms , and the TP99 (*i.e.,* Top 99 Percentiles) RT is about 36 ms.

Table 6: Statistics of some embedding tables in ZEUS.

Category	Name	Vocabulary size	Embedding size
User	age	120	4
	gender	3	4
Query	query id	100 million	32
	word segments	1 million	16
	category id	50,000	16
Product	product id	100 million	32
	word segments	1 million	16
	category id	50,000	16
	brand id	50,000	16
Scenario	scenario name	10	4

8 CONCLUSION

In this paper, we propose ZEUS, which exploits pre-training to improve the Multi-Scenario ranking problem. By performing Self-supervised Learning on Users' spontaneous Behaviors, ZEUS can alleviate the long-standing Feedback Loop problem in learning ranking models, solve the insufficient training data problem in small and new scenarios, and improve the ranking performance in all scenarios. We conduct extensive offline and online A/B Testing in Alibaba's Multi-Scenario Intent Recommeder System and demonstrate the effectiveness of ZEUS for multi-scenario ranking.

REFERENCES

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in Alibaba. In *DLP-KDD'19*. 1–4.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML'20*. PMLR, 1597–1607.

[3] Weijian Chen, Yulong Gu, Zhaochun Ren, Xiangnan He, Hongtao Xie, Tong Guo, Dawei Yin, and Yongdong Zhang. 2019. Semi-supervised user profiling with heterogeneous graph attention networks. In *IJCAI'19*. 2116–2122.

[4] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. *CVPR'21* (2021), 15750–15758.

[5] Yuting Chen, Yanshi Wang, Yabo Ni, An-Xiang Zeng, and Lanfen Lin. 2020. Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce. *ICDMW'20* (2020), 127–135.

[6] 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 *DLRS'16*. 7–10.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys'16*. 191–198.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL'19*. 4171–4186.

[9] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR'11* 12, 7 (2011).

[10] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW'15*. 278–288.

[11] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *KDD'19*. 2478–2486.

[12] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[13] Yulong Gu. 2021. Attentive Neural Point Processes for Event Forecasting. In *AAAI'21*. 7592–7600.

[14] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical User Profiling for E-commerce Recommender Systems. In *WSDM'20*. 223–231.

[15] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, Lixin Zou, Yiding Liu, and Dawei Yin. 2020. Deep Multifaceted Transformers for Multi-objective Ranking in Large-Scale E-commerce Recommender Systems. In *CIKM'20*. 2493–2500.

[16] Yulong Gu, Jiaxing Song, Weidong Liu, and Lixin Zou. 2016. HLGPS: a home location global positioning system in location-based social networks. In *ICDM'16*. 901–906.

[17] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL'20*. 8342–8360.

[18] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C Turnbull, Brendan M Collins, et al. 2019. Applying deep learning to Airbnb search. In *KDD'19*. 1927–1935.

[19] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *ADKDD'14*. 1–9.

[20] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM'18*. 667–676.

[21] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *WSDM'19*. 528–536.

[22] Pengcheng Li, Runze Li, Qing Da, An-Xiang Zeng, and Lijun Zhang. 2020. Improving Multi-Scenario Learning to Rank in E-commerce by Exploiting Task Relationships in the Label Space. In *CIKM'20*. 2605–2612.

[23] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *WSDM'20*. 331–339.

[24] Ruirui Li, Liangda Li, Xian Wu, Yunhong Zhou, and Wei Wang. 2019. Click feedback-aware query recommendation using adversarial examples. In *WWW'19*. 2978–2984.

[25] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[26] Yiding Liu, Yulong Gu, Zhuoye Ding, Junchao Gao, Ziyi Guo, Yongjun Bao, and Weipeng Yan. 2020. Decoupled Graph Convolution Network for Inferring Substitutable and Complementary Items. In *CIKM'20*. 2621–2628.

[27] 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 *KDD'18*. 1930–1939.

[28] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-Supervision in Sequential Recommenders. In *KDD'20*. 483–491.

[29] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *SIGIR'18*. 1137–1140.

[30] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *KDD'15*. 785–794.

[31] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *KDD'13*. 1222–1230.

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[33] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In *CIKM'20*. 2669–2676.

[34] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *CVPR'16*. 2536–2544.

[35] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[36] Corbin Rosset, Chenyan Xiong, Xia Song, Daniel Campos, Nick Craswell, Saurabh Tiwary, and Paul Bennett. 2020. Leading Conversational Search by Suggesting Useful Questions. In *WWW'20*. 1160–1170.

[37] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. *arXiv preprint arXiv:2101.11427* (2021).

[38] 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 *CIKM'19*. 1441–1450.

[39] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In *WWW'19*. 645–651.

[40] 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 *NIPS'17*. 5998–6008.

[41] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD'18*. 839–848.

[42] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR'21*. 726–735.

[43] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. In *SIGIR'20*. 931–940.

[44] Yatao Yang, Biyu Ma, Jun Tan, Hongbo Deng, Haikuan Huang, and Zibin Zheng. 2021. FINN: Feedback Interactive Neural Network for Intent Recommendation. In *WWW'21*. 1949–1958.

[45] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Evan Ettinger, et al. 2021. Self-supervised Learning for Large-scale Item Recommendations. *CIKM'21* (2021).

[46] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, et al. 2016. Ranking relevance in yahoo search. In *KDD'16*. 323–332.

[47] 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 *RecSys'19*. 43–51.

[48] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive Learning for Debiased Candidate Generation in Large-Scale Recommender Systems. *KDD'21* (2021).

[49] 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 *AAAI'19*. 5941–5948.

[50] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD'18*. 1059–1068.

[51] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM'20*. 1893–1902.

[52] Yu Zhu, Yu Gong, Qingwen Liu, Yingcai Ma, Wenwu Ou, Junxiong Zhu, Beidou Wang, Ziyu Guan, and Deng Cai. 2019. Query-based interactive recommendation by meta-path and adapted attention-GRU. In *CIKM'19*. 2585–2593.

[53] Lixin Zou, Long Xia, Yulong Gu, Xiangyu Zhao, Weidong Liu, Jimmy Xiangji Huang, and Dawei Yin. 2020. Neural interactive collaborative filtering. In *SIGIR'20*. 749–758.