Personalized Attraction Enhanced Sponsored Search with Multi-task Learning

Wei Zhao
State Key Lab of ISN, Xidian Univ.
ywzhao@mail.xidian.edu.cn

Boxuan Zhang
Alibaba Group
boxuan.zbx@gmail.com

Beidou Wang
Simon Fraser University
beidouw@sfu.ca

Ziyu Guan*
State Key Lab of ISN, Xidian Univ.
zyguan@xidian.edu.cn

Wanxian Guan
Alibaba Group
wanxian.gwx@alibaba-inc.com

Guang Qiu
Alibaba Group
guang.qiug@alibaba-inc.com

Wei Ning
Alibaba Group
wei.ningw@alibaba-inc.com

Jiming Chen
Zhejiang University
cjm@zju.edu.cn

Hongmin Liu
Henan Polytechnic University
hongminliu@hpu.edu.cn

ABSTRACT

We study a novel problem of sponsored search (SS) for E-Commerce platforms: how we can attract query users to click product advertisements (ads) by presenting them features of products that attract them. This not only benefits merchants and the platform, but also improves user experience. The problem is challenging due to the following reasons: (1) We need to carefully manipulate the ad content without affecting user search experience. (2) It is difficult to obtain users' explicit feedback of their preference in product features. (3) Nowadays, a great portion of the search traffic in E-Commerce platforms is from their mobile apps (e.g., nearly 90% in Taobao). The situation would get worse in the mobile setting due to limited space. We are focused on the mobile setting and propose to manipulate ad titles by adding a few selling point keywords (SPs) to attract query users. We model it as a personalized attractive SP prediction problem and carry out both large-scale offline evaluation and online A/B tests in Taobao. The contributions include: (1) We explore various exhibition schemes of SPs. (2) We propose a surrogate of user explicit feedback for SP preference. (3) We also explore multi-task learning and various additional features to boost the performance. A variant of our best model has already been deployed in Taobao, leading to a 2% increase in revenue per thousand impressions and an opt-out rate of merchants less than 4%.

KEYWORDS

Sponsored Search; E-Commerce; Multi-task Learning; Personalization

*Corresponding author

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330659

ACM Reference Format:
Wei Zhao, Boxuan Zhang, Beidou Wang, Ziyu Guan, Wanxian Guan, Guang Qiu, Wei Ning, Jiming Chen, and Hongmin Liu. 2019. Personalized Attraction Enhanced Sponsored Search with Multi-task Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330659

1 INTRODUCTION

Online shopping

Figure 1: An ad returned to a comfort-seeking user who is looking for "High-waisted jeans for women": (a) with the original title, and (b) with the refined title by our system. Texts in red bounding boxes are SPs. In the original title, SPs are hidden in the title, while in the refined one the most attractive SP is promoted to the front with emphasis.

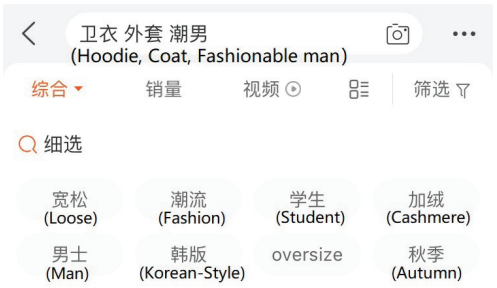

Figure 2: The search system of Taobao can provide specific feature keywords (SFs) related to the user query. Users can then click these SFs to narrow down the search space.

[14]. These findings are important for helping advertisers to refine their ads in a general sense, but are orthogonal to our problem where we need to predict which SP best attracts the query user. Researchers have also proposed solutions for word attractiveness estimation [11, 15, 22]. However, these methods all evaluated attractiveness without considering preference discrepancies among query users. In the E-Commerce context, personalization is very important since users can have quite different demands when looking for the same kind of products. Fig. 1(a) shows an example of the original product title in search results. We can see two examples SPs are hidden in the title. When looking for "High-waisted jeans for women", idolaters may prefer the first SP "Star-Style", while some other people may only be concerned about comfort and therefore we should make "Slight-Elastic" easily noticeable to them. Such inconspicuous SPs are not able to give users personalized experience, or even bypassed by them. In comparison, our system can put "Slight-Elastic" at the front with emphasis for users who like this feature (Fig. 1(b)). Providing personalized SP impressions to query users can not only benefit merchants and platforms by improving CTR, but also improve user search experience by showing their most concerned features to them. In Taobao, candidate SPs of an ad are mainly provided by the advertiser. Although Taobao also tries to automatically mine SPs, how to obtain candidate SPs is out of scope of this paper.

This automatic ad refinement problem is challenging due to the following reasons: (1) We must carefully manipulate the ad content so as to effectively attract users' attention without affecting their search experience. (2) Although user click data is abundant, it is difficult to obtain users' explicit feedback for their preference in product SPs. (3) As aforementioned, nowadays most traffic to E-Commerce platforms is from mobile devices. Due to space limitation on mobile devices, it is difficult to present many SPs to users.

In this work, we are focused on the mobile setting and study how we can manipulate the ad titles by adding a few proper SPs to improve the CTR of ads. Firstly, we explore the impact of various exhibition schemes of SPs on the CTR performance. Secondly, we design a neural model for the task of personalized attractive SP prediction given a user and his/her submitted query. To alleviate the problem of lack of explicit user preference to SPs, we propose to employ the user click information on *specific feature keywords* (SFs) as a surrogate to user explicit feedback for SP preference, and train the model accordingly. SFs are presented to a query user after he/she submits a query, to help narrow down the search space. Most E-Commerce Websites have this function. An example is shown in Fig. 2. Like SPs, these SFs also represent the specific features of products. Nevertheless, the diversity and click data of SFs are still limited. On the other hand, users' historical CTR data can implicitly reflect their preference in SPs. For instance, if we observe that a user often clicks products with "free shipping" in the title, the user would probably favor products with such a feature. Hence, we employ CTR prediction as an auxiliary task to perform multi-task learning, trying to boost the performance of the main task (i.e., personalized attractive SP prediction). Finally, we further incorporate various additional features of users and queries into our model to better characterize them. Both large-scale offline evaluation and online A/B tests are systematically carried out in the Taobao platform to verify the effectiveness of these ideas.

A variant of our best model has already been deployed in Taobao, leading to a 2% increase in revenue per thousand impressions and an opt-out rate of merchants less than 4%. Readers can experience this new function in Taobao App.

2 EFFECTIVENESS OF PERSONALIZED ATTRACTION ENHANCED SPONSORED SEARCH

In this section, we propose the first series of large-scale experiments and in-depth discussions to reveal the effectiveness of boosting the user attraction of an E-Commerce SS system by refining the ad content in search results. Using 120 million sponsored search impressions from Taobao, one of the largest E-Commerce platforms in China, online experiments and user studies are conducted to provide insights from users, merchants and the E-Commerce platform.

To be specific, we refine the *sponsored search results* (i.e., the returned records of ads) by automatically attaching personalized selling point keywords (SPs) to improve its attractiveness. In particular, experiments and analysis from this section aim to answer the following questions:

- *Can the attractiveness of SS results be improved by refining their content?*
- *Is personalization essential for the attraction enhancement?*

- *How do different factors of content refinement, including the display format of the added SPs and the number of SPs, affect the attractiveness of SS results?*

Answering these questions is of great significance. On the one hand, they are related to the fundamental assumptions of our proposed attraction enhanced SS framework; On the other hand, the investigation of these questions brings the essential insight on how to build a more attractive and intuitive E-Commerce SS system.

2.1 Effectiveness of Personalized Refinement of Sponsored Search Results

In this paper, we propose to refine the SS results and increase their attractiveness by adding personalized SPs in the front of the original SS results. Two assumptions need to be verified: 1.) adding personalized SPs helps ads to attract more user clicks than not adding; 2.) personalized exhibition of SPs helps to improve the attractiveness compared to non-personalized exhibition.

Two online A/B tests are conducted independently to verify the two assumptions. For each test, search traffic from Taobao is equally split into the control and treatment groups. Click-through rate (CTR) is used as the metric to evaluate the attractiveness of the SS results. Furthermore, we also calculate the p-value according to Fisher's exact test [23] to assess whether the results are significant. Three strategies are used to generate the SS results, including:

- **Title-based Sponsored Search result (TSS)** uses product titles provided by merchants to generate SS results and this is the traditional way to show ads, without any refinement.
- **Personalized Selling Point keywords enhanced SS result (PSPSS)** adds personalized SPs in the front of ad title in the corresponding SS result. Two SPs most attractive to the target user are selected. Attractiveness is evaluated by our basic model introduced in detail in Section 3.
- **Non-personalized Selling Point keywords enhanced SS result (NSPSS)** picks the most popular SP keywords from an ad's candidate SP set according to the click log for that ad's category (thus capturing global preference of users for that category), and puts them in the front of the ad title in the corresponding SS result.

All the above-mentioned strategies share the same text-length constraint on the SS results. The part of the title that exceeds the text-length constraint will be cut off.

2.1.1 Effectiveness of Adding Personalized SPs. As demonstrated in Experiment 1 from Table 1, compared with the traditional SS results without SPs (TSS), adding personalized SPs boosts the CTR of ads by 1.9%. It is worth noting that the SS engine in Taobao is a well established system and usually a 0.3% improvement on CTR is considered to be significant and the significance is also confirmed by the p-value. It strongly indicates adding SPs to SS results is a right direction for enhancing ad attractiveness. The effectiveness is also proved by our user study with over 95% surveyed users preferring the enhanced version of SS results. It is worth noting the personalized SP prediction model used in PSPSS is just the basic version of our proposed models. After the initial online evaluation confirms our assumptions, more advanced models are designed and empirically tested with both online and offline experiments. Details will be reported in Sections 4, 5 and 6.

There are two clear reasons behind the significant improvement. On the one hand, the added SPs reflect the products' most attractive features to the target users and inevitably help to attract users to click on the enhanced SS results. On the other hand, nowadays most of the users use their mobile devices for online shopping (90% of Taobao's search traffic comes from the mobile-end). The display space for a SS result on a mobile device is usually very limited. For instance, in Taobao app, an SS result can only use up to 28 Chinese characters for its title and users have to make their decision with the information provided in the limited space. The personalized SPs added by our model usually reflect the features that a user cares most. This greatly increases the informativeness of the SS results and thus increases click-through rate.

2.1.2 Effectiveness of Personalization. In our second A/B test, we investigate how much personalization helps to boost the attractiveness of SS results by comparing adding personalized SPs (PSPSS) vs. adding non-personalized SPs (NSPSS). As demonstrated in Experiment 2 from Table 1, a 0.55% improvement is observed on CTR with the help of personalization. The p-value of 0.0009 also indicates a significant result. Experiment 2 confirms that adding personalized SPs is the right direction for our algorithm design.

The significant improvement is expected. Compared with non-personalized SPs generated in NSPSS, the personalization of PSPSS can help the search engine better target a user by demonstrating the features that he/she is the most interested in. This is also backed up by our user study: 82% of the surveyed users point out that personalization can help them better locate the products they want.

Table 1: Effectiveness of Personalized Refinement (CTR scores reflect relative CTR change of treatment over control).

Exp. ID	Control	Treatment	CTR	P-Value	Total Impressions
1	TSS	PSPSS	+1.9%	0.00001	20,500,000
2	NSPSS	PSPSS	+0.55%	0.0009	40,874,000

2.2 Impact of SP Exhibition Factors

Besides the algorithm to pick the best personalized SPs (discussed in detail in the next sections), there are also some exhibition factors to be considered for SS results refinement. In particular: How do we emphasize the added SPs so as to attract query users? How many SPs should we add to a single SS result? These factors in fact proved to be very important for building an effective attraction enhanced E-Commerce SS system based on our large-scale A/B tests.

2.2.1 Impact of SP Emphasis. In this part, we aim to investigate whether UI-based emphasis on the added SPs will make a difference to the attractiveness of SS results.

To evaluate the impact of SP emphasis, we conduct an online A/B test to compare two types of SP display UI. The emphasized version comes with bold box brackets around each SP to highlight it and the non-emphasized version comes without the brackets

Table 2: The impact of SP emphasis.

Experiment ID	Control	Treatment	CTR Change (Treatment Over Control)	P-Value	Total Impressions
3	No Emphasis	With Emphasis	+0.39%	0.0167	20,500,000

Table 3: The impact of the number of SPs.

Experiment ID	Control	Treatment	CTR Change (Treatment Over Control)	P-Value	Total Impressions
4	2 SPs added	3 SPs added	+0.09%	0.7347	21,490,000
5	2 SPs added	1 SP added	-0.53%	0.001	20,200,000

Figure 3: The left case shows two SPs wrapped in bold box brackets which highlight them; the right case shows two SPs without emphasis.

(examples in Fig. 3). As displayed in Table 2, this type of emphasis helps to improve CTR by 0.39%, with p-value=0.0167 (significant when significance level=0.05). We also tried out parentheses, normal box brackets and square brackets, but none of them works as good as the bold box brackets.

The explanation behind the scene can be interesting. The original SS result, which directly displays the title provided by the merchant, is usually lengthy and packed with merchant-provided keywords. Users can easily get swamped by the information and ignore the added SPs, as shown by the lower case of Fig. 3. The emphasis UI helps us grab users' attention to our proposed personalized SPs, which further enhances the attractiveness of SS results and consequently improves CTR.

2.2.2 Impact of the Number of SPs. Since we can add multiple SPs to a SS result, a natural question is how many SPs we should add. Does it follow "the more, the better" rule? Two online A/B tests are conducted to investigate the impact of adding different numbers of SPs. We investigate the CTR improvements of "adding two SPs vs. adding three SPs" (Experiment 4) and "adding two SPs vs. adding one SPs" (Experiment 5). The results are shown in Table 3. Adding only one SP is significantly outperformed by adding two SPs. While the improvement of three SPs over two SPs is 0.09%, its p-value shows no significance.

The results indicate only showing one SP is not reliable. One reason could be the imperfectness of the prediction model, i.e., failing to put the SP a user likes the most at the 1st position. Although we could improve the prediction model, showing two SPs seems to be more safe and do not hurt user experience. Regarding "two SPs vs. three SPs", there are two possible explanations. First, adding too many SPs tends to be overwhelming and distract a user from noticing the SP that truly attracts him. Secondly, the display space on mobile apps is usually very limited: each SS result on the Taobao app can only display 28 Chinese characters. Adding too many SPs will force a large portion of the original product title to be cut off and lead to negative user experience.

3 THE BASIC MODEL

We have shown the importance of personalization for attractive selling point keyword (SP) prediction. In the next, we will detail the models we develop for personalized attractive SP prediction. We present firstly in this section our basic neural model for this task. This model is trained with users' click data on specific feature keywords (SFs) introduced in Section 1.

3.1 Common Notations & Formulation

We are given a set of users $\mathcal{U}$ and a set of ads $\mathcal{A}$. Let $\mathcal{W}$ denote the set of all keywords. Each ad $a \in \mathcal{A}$ has a set of SPs denoted as $\mathcal{T}_a = \{t_1, t_2, \ldots, t_m\}$, where $t_i \in \mathcal{W}$. When a user $u \in \mathcal{U}$ submits a query $q = \{w_1, w_2, \ldots, w_s\}$ ($w_i \in \mathcal{W}$), the search engine returns a set of ads. The goal is, for each returned ad a, to predict $p(t_i|u, q)$, the probability that each $t_i \in \mathcal{T}_a$ attracts u given q. Then the SPs with the highest probabilities can be added to the SS result of a in the search result list. Note the condition in $p(t_i|u, q)$ does not contain a. This is because given a submitted query, a user's preference in product features is deterministic and independent of specific ads.

In this paper, we estimate $p(t_i|u, q)$ via a neural model. Since we do not have users' explicit feedback for their preference in SPs, we employ users' click information on specific feature keywords (SFs) to train the model. Formally, we have a set of labeled SF click data $\mathcal{S}^{SF} = \{(u, q, v, y)\}$, where $v \in \mathcal{W}$ denotes a SF and y is the binary label indicating whether u clicks v[3]. The problem becomes, training a neural model for estimating $p(t_i|u, q)$ with supervision on $\mathcal{S}^{SF}$.

3.2 Model Architecture & Training

The model structure of our basic model is depicted in Fig. 4(a). At the input layer, we have three parts: user u, query q and SF v. q is a set of keywords and v is also a keyword. As to the user u, we extract his/her recent click history for representation construction. Specifically, we use the frequent keywords in a user's click history to represent his/her long-term and short-term preferences (details can be found in the appendix). The advantage of this user representation scheme is two-fold: (1) modeling users' long-term and short-term interests conveyed by keywords can achieve proper personalization in estimating $p(t_i|u, q)$; (2) the "out-of-sample" issue can be naturally handled. That is, we can easily generalize the model to users unseen in the training stage. We can generate a user's representation as long as he/she has clicked a few products.

[3]We will detail how to sample negative instances in the appendix

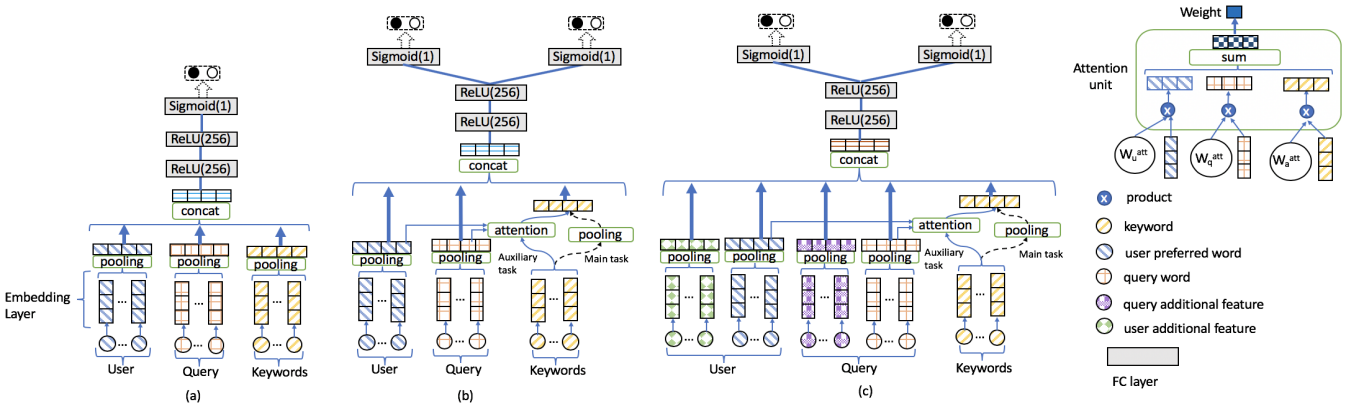

Figure 4: The three proposed models for personalized attractive SP prediction: (a) the basic model, (b) the multi-task model, and (c) the augmented model with additional features of uses and queries.

u and q are modeled as multi-hot encoding vectors based on $\mathcal{W}$. The details of the remaining layers are as follows:

- **Embedding Layer**: The keyword set $\mathcal{W}$ is typically a very large set, resulting in very high dimensionality of multi-hot vectors. A popular way for deep learning models to reduce the dimensionality of multi-hot encoding vectors is to add embedding layers [18], which transform them into low dimensional dense vectors via an embedding table. Let $\mathbf{Em}(w)$ be a look-up function that returns the embedding vector for keyword w. The embedding layer transforms u, q and v into the corresponding embedding representations. For example, q becomes $\{\mathbf{Em}(w_1), \mathbf{Em}(w_2), \ldots, \mathbf{Em}(w_s)\}$.
- **Pooling Layer & Concatenation Layer**: The Embedding Layer transforms u and q into variable-size sets of vectors. To obtain fixed-length representations to facilitate further computation, we add a Pooling Layer above the Embedding Layer (average pooling is used in this work, e.g., for query: $\mathbf{e}_q = \frac{1}{s}\sum_{i=1}^{s}\mathbf{Em}(w_i)$). Note that a pooling operation is also placed over the SF embedding to make the model general. This is because a few SFs/SPs may contain multiple keywords. However, we still call them keywords for clarity and simplicity. After pooling, the Concatenation Layer simply synthesizes information from the three channels: $\mathbf{x} = [\mathbf{e}_u^T, \mathbf{e}_q^T, \mathbf{e}_v^T]^T$.
- **Fully-connected Layers & Output Layer**: We then feed the obtained $\mathbf{x}$ through two Fully-connected (FC) Layers with the RELU nonlinear activation function [16] to increase the expressiveness of the model. The final Output Layer is simply a logistic regression for label prediction.

The training protocol for the basic model is standard supervised training with the cross-entropy loss.

What we want to point out is that, this basic model is simple without in-depth techniques. However, we choose to stick to this simple design since: (1) the model should be able to generate real-time predictions when deployed in Taobao's search engine; (2) the major focus of this work is to investigate whether and how we can exploit user behaviors (click information on products and SFs) and rich features for personalized attractive SP prediction. We leave possible improvements for the model to future work.

4 THE MULTI-TASK MODEL

This section details how we use the CTR prediction task to boost the performance of the personalized attractive SP prediction task. We refer to them as *auxiliary task* and *main task*, respectively.

4.1 Why using CTR prediction?

Although users' click data on SFs can be regarded as providing direct supervision for the SP prediction task, the available information is still limited in the following two aspects: (1) the number of SF clicks is limited. In product search, users are more likely to click a returned product record than a shown SF; (2) the set of all SFs is limited and only covers a subset of the set of SPs.

In comparison to clicks on SFs, there are much more clicks on product records in product search and the product titles cover more diverse SPs[4]. Users' product click information can be readily obtained from the search engine log. Although product click data cannot provide direct supervision signals for the SP prediction task, they do imply to some extent how a user is attracted by the returned products in the context of the corresponding queries. For example, if we often observe that a user clicks products with the keyword "Slight-Elastic" in the title when searching for "Jeans", it is reasonable to believe the user likes jeans with slight elasticity as a feature. Hence, click data can be regarded as providing weak/implicit supervision for our main task. Though most of the time users are attracted by feature keywords in product titles, sometimes users may also be attracted by other factors of the returned product records, such as pictures. To reduce the impact of other factors, we collect only the click data for ads equipped with SP exhibition (generated by the basic model). When looking at these ads, users are more likely to be attracted by their titles due to the attractive presentation style (see Fig. 3). Formally, we define $\mathcal{S}^C = \{(u, q, a, y^c)\}$ to be a set of labeled user ad click data, where $a \in \mathcal{A}$ denotes an ad and y^c is the binary label indicating whether u clicks a.

4.2 Model Architecture

The multi-task model is shown in Fig.4(b). As can be seen, we take a hard parameter sharing scheme [21]. The overall structure is similar

[4]The Merchants always try to increase their products' visibility and CTR by adding exhaustive feature keywords in the title.

to the basic model. The only two differences are: (1) the output layers are separate for the two different tasks; (2) the input units for SPs (SFs) also accept ad titles, but we substitute an attention module for the pooling function at the pooling layer for the auxiliary task. The intuition for incorporating attention is that, since product titles typically contain much more keywords than SPs/SFs, an attention module can help distill the most important keywords explaining the corresponding click and also make the output distributions at the pooling layer more compatible between the two tasks. Let $\mathcal{D}_a = \{d_1, d_2, \ldots, d_n\}$ denote the set of keywords in title $\mathcal{D}_a$ of ad a. We adopt a popular attention mechanism which is specified as follows

$$b_j = \mathbf{z}^T \tanh(\mathbf{W}_u^{att}\mathbf{e}_u + \mathbf{W}_q^{att}\mathbf{e}_q + \mathbf{W}_a^{att}\mathbf{Em}(d_j))$$
$$\alpha_j = \frac{\exp(b_j)}{\sum_{i=1}^{n}\exp(b_i)}, \quad \mathbf{e}_a = \sum_{j=1}^{n}\alpha_j \mathbf{Em}(d_j)$$

Here we estimate the attention score of each word in $\mathcal{D}_a$ with respect to both u and q, to obtain a personalized result.

We believe that whether a user clicks an ad with SP exhibition is highly correlated with the attractiveness of the exhibited SPs to him/her. Hence, the two tasks are well correlated. By hard parameter sharing, useful knowledge from the auxiliary task could be transferred to the main task by the parameters for mapping users, queries and SPs/ad titles into high-level representations, thus facilitating the SP prediction problem. The training details of this model can be found in the appendix.

5 THE AUGMENTED MODEL

In this section, we present the augmented model equipped with both multi-task learning and additional features of the concerned user and query.

5.1 Incorporating Features into the Model

The augmented model is the same as the multi-task model, except for the input layer, where we add additional features for users and queries to better characterize them (green and purple nodes in Fig. 4). In our framework, features are represented in a multi-group mutli-hot encoding form. Each group contains multiple discrete categorical features or bag-of-words (BoW) features that are semantically related. For example, suppose we have three features for users: *gender*, *occupation* and *preferred brands*. The former two are discrete categorical features, while the last one is a BoW feature. These features form two groups: {gender, occupation} (user profile) and {preferred brands} (user preference). For a user with feature values [gender=male, occupation=doctor, prefer brand={Nike,Adidas}], the corresponding multi-group mutli-hot encoding vector could be:

$$[\ \{\ \underbrace{1, 0}_{gender}\ ;\ \underbrace{0, \ldots, 1, \ldots, 0}_{occupation}\ \}\ \{\ \underbrace{0, \ldots, 1, \ldots, 1 \ldots, 0}_{preferred\ brands}\ \}\ ]$$

where we use "{}" as group delimiters and use ";" to separate features within the same group. This multi-group mutli-hot encoding vector is then fed to the embedding layer and the pooling layer to generate fixed-length vectors, as in the basic model. Here the pooling operation is performed within each group.

5.2 Features

Table 4 shows all the additional features used in our framework. In case a feature is numerical or continuous, it is discretized. We discuss these added features group by group. Detailed information of these features can be found in the appendix.

User profile information. This group includes users' demographic features. Compared to using the long- and short-term interests of users only, these features further characterize users in a fine-grained level. Users with different demographic features may show different preferences in product features even if their interests are similar. For example, among people who like wearing jeans, young people may prefer the SP "Star-style" while old ones care more about comfort. Thus, demographic features could help the model better capture a user's preference in SPs.

User general preference. This group of features encodes a user's preference regarding some important general aspects: product categories, product brands and whether the user likes discounts. These features could be useful for both tasks: for the auxiliary task, the general preference of a user accounts for the bias of user behaviors, which could better help the model correctly explain clicks so that more accurate representations could be transferred to the main task; for the main task, these features further enrich user preference and could help better evaluate personalized attractiveness. For instance, we could present brand-related (or discount-related) SPs to a user if we know the user has preferred brands (or likes discounts).

User consumption/activity level. these features measure the consumption level and/or activity level of a user from different angles. Users with different consumption/activity levels may have different behavioral biases and tastes. For example, a user with high consumption level could care less about price but pay more attention on quality. Incorporating these features could also benefit both the two tasks.

Query category. For queries, we add their category information (estimated by another module in the search engine) as a BoW feature to provide contextual information. The intuition is that, users generally pay attention to different features for products of different categories. For instance, people often concentrate on performance for PC, while for laptops weight is usually more important. If we only have a user's historical click information for the PC category, the basic and multi-task models may wrongly rank SPs related to performance higher when the user searches for laptops.

6 EXPERIMENTS

Based on large-scale online experiments in Section 2, it has already been proved that adding personalized selling point keywords (SPs) to Sponsored Search (SS) results leads to more attractive SS results compared with not adding SPs or adding non-personalized SPs. In this section, both online and offline experiments are conducted to compare the personalized SP prediction models proposed in Section 3-5. We also analyze important factors that impact the performance of these models. In particular, we aim to answer the following questions: (1) How well do different versions of our proposed personalized SP prediction neural model work? (2) Will multi-task learning help to improve the performance? (3) Are the additional

Table 4: A summarization of additional features used.

Entity	Feature	Description
User	gender, age, occupation city, province	user profile information
	preference for categories preference for brands preference for discount	user general preference
	purchase level, VIP level high consumption visitors top class visitors	user consumption/ activity level
Query	category	query category

Table 5: The statistics of datasets.

	SF dataset	AD dataset
total # of clicks	47,356,924	290,336,555
total # of users	7,946,738	
% of users with over 10 clicks	12.24%	76.13%

features introduced in Section 5 useful? (4) How do different training strategies (see Section B of the appendix) affect the performance of the multi-task model?

6.1 Datasets

In collaboration with Alibaba, our datasets are collected from Taobao, its C2C platform with over 500 million monthly active users. For training and offline evaluation purpose, we collect over 7 million users' click-through records for click information on specific feature keywords (called SF dataset) and on product search advertisements (called AD dataset). Statistics details of the collected SF and AD datasets are summarized in Table 5. The SF dataset is used to train our basic model and also serves as the input of the main task for our multi-task learning models. The AD dataset is used as the input of the auxiliary task of our multi-task learning models. It is worth noting that, as displayed in Table 5, we also calculate the percentage of users with over 10 clicks in the SF and AD datasets. AD dataset has 6 times of users with over 10 clicks compared with SF data, which makes AD a good auxiliary data source for our main task. Details of processing of the two datasets are in the appendix.

6.2 Compared Algorithms

Three algorithms corresponding to the three models proposed in the above are compared in this section: **The Basic Model** is proposed in Section 3 and is trained only using users' click data on specific feature keywords (SFs); **The Multi-task Model** is proposed in Section 4 which uses both SF data and AD data at the same time; **The Augmented Model** is proposed in Section 5 which extends the multi-task model by adding additional features.

The Area Under the receiver operating characteristic Curve (AUC) [3] is used as the evaluation metric for offline experiments and click-through rate (CTR) is used as the metric for online experiments. As in Section 2, p-value for Fisher's exact test is calculated to assess the significance of improvements in online experiments.

6.3 Comparison Analysis

We compare the three proposed models with both offline and online experiments. In the offline setting, the SF dataset is considered to be the ground truth representing users' preference on SPs and the three models are evaluated on the holdout test set. And in the online setting, we deploy the three models in Taobao's search engine and use large-scale online A/B tests to compare them. Totally over 90 million search impressions are collected in our A/B tests and in each A/B test, traffic is equally divided into the control and treatment groups. The results are reported in Tables 6 and 7 respectively. From Table 7, We can see both the augmented model and the multi-task model significantly outperform the basic model, with significance level 0.05. The augmented model shows the best performance. These results are consistent with the offline results.

These results are within our expectation. The multi-task model beats the basic model which is only trained on SF data. The reasons should be that, the volume of the SF data is limited and the set of SF keywords can only cover a subset of SP keywords. By introducing multi-task learning with CTR prediction for ads as an auxiliary task, extra useful information on users' implicit preference towards SPs is transferred to the personalized SP prediction task and thus improves its performance.

The augmented model exceeding the multi-task model reveals another important insight in designing an effective SP prediction model: besides the model itself, the features used in the model also play an important part in improving its effectiveness.

6.4 Dissection of the Multi-task Model Results

As mentioned in Section 6.3, both offline and online experiments confirm that multi-task learning helps to improve the performance of SP prediction by introducing auxiliary knowledge from a user's ad click history. In this section, we analyze the results of the multi-task model in detail to find out what kinds of users can benefit more from multi-task learning.

We divide instances from the test set of SF dataset into four groups based on the corresponding users' numbers of clicks for main task and auxiliary task in the training data. That's to say, we try to find out whether a user has abundant information in the training data of main and auxiliary tasks will affect the performance gain of the multi-task model over the basic model. Due to the power-law effect, we choose the median of the number of user clicks in the main/auxiliary task training data as the threshold for dividing user. The results are summarized in Table 8.

Based on the results, the highest performance gain is obtained for users with insufficient data in the main task but with abundant data in the auxiliary task (the first line). This makes sense because our proposed multi-task model aims to solve the data sparsity problem in the main task by transferring knowledge from the auxiliary task. Similarly, results from Table 8 also confirm that the lowest performance gain is observed for users who have abundant data in the main task but insufficient data in the auxiliary task. For them, we still can get a 2.99% improvement.

6.5 Feature Ablation Study

As confirmed in Section 6.3, features play an important role in improving SP prediction performance. In Section 5.2, four groups

Table 6: Offline comparison results.

Model	AUC
The Basic Model	0.5995
The Multi-Task Model	0.6215 (+3.67%)
The Augmented Model	0.6261 (+4.44%)

of features were proposed to be added as additional features. In this section, we conduct an ablation study of those features. Specifically, we perform a series of offline experiments by adding these groups of features one at a time and evaluate the effectiveness of each group of features based on the gain of AUC. As shown in Table 9, all the four groups of features proposed in Section 5.2 bring useful information for personalized SP prediction. When integrated with all of the four groups, our model achieves the best performance, indicating they are generally complementary to one another.

6.6 Influence of Different Training Strategies

Two training strategies, pre-training and alternative training, are introduced in Section B of the appendix. In this section we compare these two training strategies in the offline experimental setting.

The results are presented in Table 10. We find that the pre-training strategy performs worse than the alternate training strategy. The main reason behind this could be that in alternate training, the main task interacts and optimizes together with the auxiliary task continuously, while in pre-training the auxiliary task only serves as a prior for the main task, without much interaction.

6.7 Online System Deployment

Serving over 500 million active users is not a trivial task. In deployment, we encountered some challenges. One of the main concern in designing our algorithm is to balance between complexity of the model and computational efficiency. Taobao has strict constraints on the response time of algorithms deployed in production. In order to make real-time predictions with low latency, we accelerated online serving of industrial deep networks through AVX (Advanced Vector Extensions) instruction set supported by Intel CPUs. Thanks to it, the computational performance increased by about 7 times. Based on a large-scale online analysis, our proposed framework on average can generate personalized SPs for hundreds of SS results for a single user in less than 5 milliseconds, which supports a swift user experience. Second, after the system being in production for a while, we received several complaints about bad SPs from merchants. In response, we recalled these bad cases for analysis. Then we designed artificial rules to post-process the automatically mined SPs, to make them intuitive and easy to understand (e.g., from "old man" to "for old man").

Table 7: Results for online comparisons.

Control	Treatment	CTR	P-Value	Total Impressions
Basic Model	Multi-Task Model	+0.24%	0.0273	46,780,000
Basic Model	Augmented Model	+0.49%	0.0025	46,200,000

Table 8: Performance of the multi-task model w.r.t different groups of users divided according to the numbers of clicks in the training data of main/auxiliary tasks.

Main Task	Auxiliary Task	AUC of Basic Model	AUC of Multi-task Model
clicks<=6	clicks>21	0.6107	0.6341 (+3.83%)
clicks<=6	clicks<=21	0.6075	0.6297 (+3.65%)
clicks>6	clicks>21	0.5906	0.6113 (+3.50%)
clicks>6	clicks<=21	0.5850	0.6025 (+2.99%)

Table 9: Ablation study for features used in the augmented model.

Description	AUC (Gain)
No additional features	0.6215
Only add user profile information	0.6234 (+0.31%)
Only add user general preference	0.6225 (+0.16%)
Only add user consumption/activity level	0.6226 (+0.18%)
Only add query category	0.6246 (+0.50%)
Add all additional features	0.6261 (+0.74%)

Table 10: AUCs of different training strategies for multi-task learning.

Training strategies	AUC
Alternate training	0.6215
Pre-training	0.6132

7 RELATED WORK

Sponsored search (SS) has achieved great industrial success. It is the major business model for most online search service providers. In academia, SS has also attracted a lot of attention from researchers in related fields. Typical research problems for SS include bidding optimization [2, 4, 8, 28], click prediction [1, 9, 12, 24], keyword suggestion [20, 25], auction mechanism design [10], etc. However, the majority of existing works were focused on the platform side or the advertiser side. Only a few research works have considered the attraction of ads to users. Wang *et al.* [24] studied why users clicked ads from the Psychological aspect. They used Maslow's desire theory to model user psychological desire in SS and automatically mined textual patterns in ads triggering users' desires. Haans *et al.* [14] empirically explored the impact of evidence type on CTR and conversion rate based on Google AdWords. Kim *et al.* [15] developed an advertiser-centric click prediction approach for mining attractive words from ad texts which can then be suggested to the advertisers for ad refinement. Govindaraj *et al.* [11] used the estimated word occurring probabilities in clicked documents to assess attractiveness. Our work is different from them in two aspects: (1) the above studies were focused on the general SS setting, while we are concerned with attractive SP prediction in E-Commerce SS. No existing methods can be applied to solve this problem. (2) More importantly, none of the above works considered personalization which is important in the E-Commerce setting. Thomaidou *et al.* [22] proposed an heuristic solution for attractive

ad description generation in the E-Commerce setting. However, the proposed method is not applicable to our problem and it did not consider personalization either.

Our work is also related to Multi-task learning (MTL) [5]. MTL aims to share useful information among multiple tasks to improve the model performance for each task. The motivation of MTL is that human beings often apply the knowledge learned from previous tasks to help learn a new task. Hence, MTL can be regarded as a form of transfer learning [19], and is also related to other areas in machine learning such as multi-label learning [26]. Readers can refer to some recent surveys [21, 27] for a complete view of MTL. Recently, deep learning based MTL methods have achieved promising performance [21]. The general idea is to share model (sub-)structures among different tasks. Two popular sharing schemes are hard parameter sharing [17] and soft parameter sharing [7]. Hard sharing means multiple tasks share the same copy of model (sub-)structure; soft sharing keeps separate models for different tasks and uses regularization to encourage the parameters to be similar. Our multi-task model takes the hard sharing scheme, since we believe the auxiliary task (CTR prediction for ads equipped with SP exhibition) is well correlated with the main task (attractive SP prediction). That is, whether a user clicks an ad with SP exhibition is highly correlated with the attractiveness of the exhibited SPs to him/her. Our multi-task model is slightly different from the standard hard sharing model in that, (1) the attention module is exclusive for the auxiliary task; (2) the two tasks do not share the same input, i.e., (u, q, v) vs. (u, q, a). Hence, the multi-task model is a customized MTL model for our problem.

8 CONCLUSION

In this paper, we studied a novel problem for E-Commerce sponsored search: enhancing the attractiveness of SS results by adding personalized attractive selling point keywords to them. We systematically carried out online A/B tests in Taobao to verify the feasibility of this idea and also explored proper schemes for SP exhibition. In addition, we tried to attack the problem of training an effective model for this task from three aspects: (1) using users' click data on specific feature keywords as a surrogate of users' explicit preference on SPs for model training; (2) incorporating the CTR prediction task as an auxiliary task to perform multi-task learning, in order to boost the performance of personalized SP prediction; (3) incorporating additional features of users and queries to further improve the performance. Large-scale offline and online experiments confirmed the effectiveness of these ideas.

ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61672409, 61522206, 61876144, 61876145), the Science and Technology Plan Program in Shaanxi Province of China (Grant No. 2017KJXX-80), the Fundamental Research Funds for the Central Universities (Grant Nos. JB190301, JB190305), and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies. The authors would like to thank the colleagues of Alimama for their supports, including Jinping Gou, Changyou Xu, Xinyang Guo, Sheng Xu, Jing Chen, Qian Wang and Ju Huang.

REFERENCES

[1] Josh Attenberg, Sandeep Pandey, and Torsten Suel. 2009. Modeling and predicting user behavior in sponsored search. In *SIGKDD*. ACM, 1067–1076.
[2] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. In *WWW*. ACM, 531–540.
[3] Andrew P Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30, 7 (1997), 1145–1159.
[4] Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, George Mavromatis, and Alex Smola. 2011. Bid generation for advanced match in sponsored search. In *WSDM*. ACM, 515–524.
[5] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.
[6] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12, Jul (2011), 2121–2159.
[7] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL-IJCNLP*, Vol. 2. 845–850.
[8] Jon Feldman, S Muthukrishnan, Martin Pal, and Cliff Stein. 2007. Budget optimization in search-based advertising auctions. In *ACM EC*. ACM, 40–49.
[9] Hongchang Gao, Deguang Kong, Miao Lu, Xiao Bai, and Jian Yang. 2018. Attention Convolutional Neural Network for Advertiser-level Click-through Rate Forecasting. In *WWW*. International World Wide Web Conferences Steering Committee, 1855–1864.
[10] Nicola Gatti, Alessandro Lazaric, and Francesco Trovò. 2012. A truthful learning mechanism for multi-slot sponsored search auctions with externalities. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, 1325–1326.
[11] Dinesh Govindaraj, Tao Wang, and SVN Vishwanathan. 2014. Modeling attractiveness and multiple clicks in sponsored search results. *arXiv preprint arXiv:1401.0255* (2014).
[12] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. 2010. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's bing search engine. In *ICML*. Omnipress, 13–20.
[13] Alibaba Group. 2018. Alibaba Group Announces March Quarter 2018 Results and Full Fiscal Year 2018 Results. https://www.alibabagroup.com/en/news/press_pdf/p180504.pdf.
[14] Hans Haans, Néomie Raassens, and Roel van Hout. 2013. Search engine advertisements: The impact of advertising statements on click-through and conversion rates. *Marketing Letters* 24, 2 (2013), 151–163.
[15] Sungchul Kim, Tao Qin, Tie-Yan Liu, and Hwanjo Yu. 2014. Advertiser-centric approach to understand user click behavior in sponsored search. *Information Sciences* 276 (2014), 242–254.
[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.
[17] Mingsheng Long and Jianmin Wang. 2015. Learning multiple tasks with deep relationship networks. *CoRR, abs/1506.02117* 3 (2015).
[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
[19] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE TKDE* 22, 10 (2010), 1345–1359.
[20] Sujith Ravi, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, Sandeep Pandey, and Bo Pang. 2010. Automatic generation of bid phrases for online advertising. In *WSDM*. ACM, 341–350.
[21] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).
[22] Stamatina Thomaidou, Ismini Lourentzou, Panagiotis Katsivelis-Perakis, and Michalis Vazirgiannis. 2013. Automated snippet generation for online advertising. In *CIKM*. ACM, 1841–1844.
[23] Graham JG Upton. 1992. Fisher's exact test. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* (1992), 395–402.
[24] Taifeng Wang, Jiang Bian, Shusen Liu, Yuyu Zhang, and Tie-Yan Liu. 2013. Psychological advertising: exploring user psychology for click prediction in sponsored search. In *SIGKDD*. ACM, 563–571.
[25] Hao Wu, Guang Qiu, Xiaofei He, Yuan Shi, Mingcheng Qu, Jing Shen, Jiajun Bu, and Chun Chen. 2009. Advertising keyword generation using active learning. In *WWW*. ACM, 1095–1096.
[26] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE TKDE* 26, 8 (2014), 1819–1837.
[27] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).
[28] Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. 2018. Deep Reinforcement Learning for Sponsored Search Real-time Bidding. *arXiv preprint arXiv:1803.00259* (2018).

A REPRODUCIBILITY

In this appendix, we provide details of our implementation and experimental setup to help reproduce the findings in this work. We cannot release the codes and datasets due to business secret and privacy issues. However, the proposed models are rather standard without sophisticated techniques, so we believe it is easy to re-implement them with the information here.

B TRAINING DETAILS OF THE MULTI-TASK MODEL

The objective functions of the two tasks (personalized SP prediction and CTR prediction) are both cross-entropy loss:

$$\mathcal{L}_{aux} = -\frac{1}{|\mathcal{S}^C|}\sum[y^c \log(y^c)' + (1-y^c)\log(1-(y^c)')] \quad (1)$$

$$\mathcal{L}_{main} = -\frac{1}{|\mathcal{S}^{SF}|}\sum[y \log y' + (1-y)log(1-y')] \quad (2)$$

where $\mathcal{L}_{aux}$ and $\mathcal{L}_{main}$ denote the loss functions for the auxiliary task and the main task, respectively, and $y'/(y^c)'$ represents the predicted label for an instance in $\mathcal{S}^{SF}/\mathcal{S}^C$.

The training of the model is slightly different from the classic multi-task models for which joint training can be performed. By joint training, we mean the classic multi-task models can do model updating jointly for different tasks based on the same mini-batch of instances. This is because the tasks share the same input. I.e., the same instance generates different outputs for different tasks. However, our two tasks do not share the same input: (u, q, v) vs. (u, q, a). For one instance from either $\mathcal{S}^{SF}$ or $\mathcal{S}^C$, we cannot do updating for the two tasks simultaneously. Hence, we explore two training strategies, *alternate training* and *pre-training*, for model training.

The alternate training algorithm is shown in Algorithm 1. We use Θ_1/Θ_2 to denote the set of parameters related to the auxiliary/main task. The iterative procedure processes the training data by sampling mini-batches. In each iteration, the main task or the auxiliary task is selected randomly, with a probability proportion of $1 : k$. Then the related parameters are updated by picking a mini-batch from the corresponding dataset, i.e. $\mathcal{S}^{SF}$ or $\mathcal{S}^C$. We set the probability of choosing the auxiliary task k times of that for the main task. The reason is that $\mathcal{S}^C$ is typically larger than $\mathcal{S}^{SF}$. k is set so that the two datasets are processed approximately the same number of times when the algorithm stops. This assures the model is sufficiently trained on both datasets.

The pre-training strategy intrinsically performs transfer learning from the auxiliary task to the main task. As shown in Algorithm 2, we first train the model with respect to the auxiliary task using $\mathcal{S}^C$, to obtain learned Θ_1. Then we incorporate these learned parameters (except the output layer and the attention module) as a prior and further train the model with respect to the main task. We will investigate the two training schemes in the experiments.

Algorithm 1: Alternate Training

```
Initialize model parameters randomly. Θ1 is the set of
  parameters related to the auxiliary task and Θ2 is the set of
  parameters related to the main task
for iteration in 1, 2, ... do
    Choose the main task or the auxiliary task with a
      probability proportion of 1 : k
    if task is auxiliary then
        Pick a mini-batch from S^C
        Compute loss L_aux(Θ1) according to Eq.(1)
        Compute gradient: ∇L_aux(Θ1)
        Update model: Θ1 = Θ1 − ε∇L_aux(Θ1)
    end
    else
        Pick a mini-batch from S^SF
        Compute loss L_main(Θ2) according to Eq.(2)
        Compute gradient: ∇L_main(Θ2)
        Update model: Θ2 = Θ2 − ε∇L_main(Θ2)
    end
end
```

Algorithm 2: Pre-training

```
Initialize model parameters randomly. Θ1 is the set of
  parameters related to the auxiliary task
for iteration in 1, 2, ... do
    Pick a mini-batch from S^C
    Compute loss L_aux(Θ1) according to Eq.(1)
    Compute gradient: ∇L_aux(Θ1)
    Update model: Θ1 = Θ1 − ε∇L_aux(Θ1)
end
Initialize shared model parameters with Θ1 trained above.
  Initialize parameters in the output layer randomly. Θ2 is the
  set of parameters related to the main task
for iteration in 1, 2, ... do
    Pick a mini-batch from S^SF
    Compute loss L_main(Θ2) according to Eq.(2)
    Compute gradient: ∇L_main(Θ2)
    Update model: Θ2 = Θ2 − ε∇L_main(Θ2)
end
```

C IMPLEMENTATION DETAILS

C.1 Hyper-parameters

The settings of hyper-parameters are as follows: (1) Embedding layers are randomly initialized using a normal distribution around 0 with standard deviation 10^{-3} for each group. For the other layers, parameters are initialized with a uniform distribution in the range $0.036 * (-sqrt(3)/sqrt(dim), sqrt(3)/sqrt(dim))$, where dim is the size of the input. (2) The embedding sizes of keywords and features are set to 50 and 24 respectively. (3) The mini-batch size is set to 256 and AdaGrad [6] is used as the optimizer, with learning rate set to 0.03. (4) The layer sizes of the two FC layers are set as 256-256. (5) In the training stage, for both pre-training and alternate training the max numbers of epochs are set to 6 and 15 for auxiliary task and main task respectively to avoid overfitting. For the alternate training strategy shown in Algorithm 1, we empirically set the probability proportion parameter $k = 4$ which leads to a good performance for the main task. For alternate training, the training will stop when

the max number of epochs of either task is reached. The above parameters are set so as to balance the impact of main/auxiliary task, and also make sure in alternate training the model is sufficiently trained with the main task.

C.2 Hardware and Software

The proposed models are implemented on TensorFlowRS (TFRS) provided by Alibaba. TFRS is a distributed deep learning platform based on TensorFlow 1.7 used internally in Alibaba. In experiments, the trainable parameters are distributed on 200 workers (20 CPU cores for each worker) and updated asynchronously.

D DATASET PROCESSING

SF Dataset: For the main task, we construct the SF dataset from TaoBao App. As shown in Fig. 2 in the main text, these SFs are displayed for query users. Because the SF dataset contains only one-class implicit feedbacks (i.e., only users' click actions as positive feedbacks), negative sampling is required for obtaining a certain number of un-clicked SFs as negative feedbacks. We randomly sample from the un-clicked SFs to generate negative feedbacks. The ratio of positive feedbacks to negative feedbacks is kept to 1:2, which is determined by cross validation in preliminary offline experiments. For offline evaluation, the SF dataset is randomly split into training and test sets with the ratio of 9:1.

AD Dataset: For the auxiliary task, we also construct the AD dataset from TaoBao APP, in a similar fashion with the SF dataset. Specifically, we treat the clicked ads as positive feedbacks and sample un-clicked ads before the last clicked ad in a search session as negative feedbacks. In order to reduce the impact of other factors of ads, we collect only the click data for ads equipped with SP exhibition (generated by the basic model). The ratio of positive feedbacks to negative feedbacks for the AD dataset is kept to 1:6.

For both datasets, we also collect rich information about users and queries for feature. These features represent user preference in many ways.

E DETAILS OF FEATURES

The Basic Features: The basic features for users are user preferred words. For each user, we extract his/her click history within the recent 1 month for representation construction. Specifically, we collect keywords from the titles of those products that a user have clicked and use (at most) top 10 frequent keywords to represent the long-term interest of that user; we also extract (at most) top 10 frequent keywords from a user's product click data within the recent one week to represent his/her short-term interest.The basic features for query, ad and SF are query keywords, ad titles and SF keywords, respectively.

The Additional Features: There are four groups of additional features, 3 groups for users and 1 group for queries (Table 4 in the main text). Here we describe them in details.

- **User profile information:** this group contains users' demographic features, i.e. gender, age, occupation and home address. *Gender* is a binary feature. *Age* is a numerical feature, so we discretize it into 10 discrete states, i.e. $[1, 10]$, $[11, 20]$, ..., $[91, 100]$. *Occupation* is a categorical feature with about 140 occupations in Taobao. *City* and *Province* are also categorical features containing cities and provinces in China.
- **User general preference:** this group measures a user's preference regarding general aspects. *Preference for categories* and *Preference for brands* are BoW features. For the former one, we collect ads that a user have clicked, collected or bought, and count the number of ads for each category. Finally, we use (at most) top 10 frequent categories to represent that user's preference for categories. There are totally about 40,000 categories in the dataset. The feature generation process for the latter one is the same. There are about 20,000 brands in the dataset. Finally, *Preference for discount* is a binary feature indicating whether the user likes discounts.
- **User consumption/activity level:** *Purchase level* and *Vip level* are both discrete feature with 7 states used in Taobao internally. Purchase level is estimated from and positively correlated with a user's recent and historical consumptions. VIP level considers not only consumption level, but also activity level (e.g., frequency of writing reviews) of a user. *High consumption visitors* and *top class visitors* are binary variables indicating top users. High consumption means a user achieves a very high consumption level; "top class" is a service in Taobao that can be bought by high consumption users.
- **Query category:** The *Category* feature is a BoW feature. In the search engine of Taobao, a query is matched to a specific category (in case an ambiguity exists, we take all the matched categories) in a hierarchy. We employ the matched category and its ancestors as a query's category information. In the dataset, there are about 400,000 categories.