Highlights

Deep Clicking Interest Network for Reranking Hotels

Denghao Ma,Hongbin Pei,Yueguo Chen,Peng Bao,Liang Shen

- We study interest modelling for reranking hotels and identify two key challenges.
- Deep click concept is formally defined to model the multi-view semantics of a click.
- Mutual attention calibration unit is designed to calibrate the embedding of a click.
- Multi-attention aggregation unit estimates the weights of clicks from multiple views.
- Both offline and online experiments show our model outperforms the SOTA baselines.

Deep Clicking Interest Network for Reranking Hotels

Denghao Ma[a,*,1], Hongbin Pei[b,*], Yueguo Chen[c], Peng Bao[d] and Liang Shen[a]

[a]Meituan, Beijing, 100102, China
[b]Xi'an Jiaotong University, Xi'an, 710049, China
[c]Renmin University of China, Beijing, 100872, China
[d]Beijing Jiaotong University, Beijing, 100044, China

ARTICLE INFO

Keywords:
Recommendation system
Hotel search
Interest modelling

ABSTRACT

Nowadays, e-commerce platforms of hotels have become a new trend to help people book hotels online, and more and more researchers pay attention to hotel search. Although interest modeling has achieved noticeable successes in product recommendations, it has not been explored in hotel search, especially in the reranking stage. This paper studies the interest modeling task for reranking hotels, and identifies two key challenges. First, hotels are not daily necessities for most people, and thus the behaviors of a user on a hotel booking platform are usually very infrequent— high behavior sparsity challenge. Second, user's current interests are somewhat different from the user's historical interests— interest gap. Therefore, using historical clicks stemming from historical interests to model the current interests inevitably suffers the interest gap challenge. To address the challenges, we propose the deep clicking interest network (DCIN). High behavior sparsity keeps us from extracting rich semantics to characterize a user's interest preference. Accordingly, we propose the deep click concept to model the multi-view semantics of a click and then enrich the semantics of both a click and a click sequence, so that the difficulty caused by the high behavior sparsity can be addressed. To address the interest gap challenge, DCIN models users' interests with two cascaded units: (i) Mutual-attention Interest Calibration Unit (ICU), which uses the candidate hotel from the ranking stage to calibrate the embedding of every click, so that the gap between the calibrated embedding and the user's current interests can be diminished; (ii) Multi-attention Interest Aggregation Unit (IAU), which estimates the weight of a calibrated embedding from the context, user's feedback and interest consistency perspectives. So the calibrated embeddings, which are relevant to the current query context and important to the user, dominate user's current interests, and thus the interest gap can be further diminished. Extensive offline experiments are performed on a public dataset and a large-scale industrial dataset, and online A/B testing is conducted over the e-commerce platform of Meituan-Hotel. The experimental results show DCIN significantly outperforms the baselines and the improved baselines by using deep clicks. Notably, DCIN has been deployed in Meituan-Hotel, and achieved 2.40%/1.09% CTCVR/CTR elevations.

1. Introduction

Nowadays, e-commerce platforms of hotels have become a new trend to help people book hotels online, such as *Airbnb*, *Ctrip*, and *Meituan-Hotel*. Take *Meituan-Hotel* as an example, the biggest online hotel booking platform in China, it sold more than 350 million room nights in 2020[2]. Because of the importance of hotel e-commerce platforms, more and more studies Abdool, Haldar, Ramanathan, Sax, Zhang, Manaswala, Yang, Turnbull, Zhang and Legrand (2020); Antognini and Faltings (2020); Ludewig and Jannach (2019) pay attention to the hotel search task. Using hotel e-commerce platforms, a user is instructed to enter a check-in location, check-in and checkout dates. So hotel search takes the check-in location, the check-in and checkout dates as input and returns a ranked list of hotels as output. In a typical pipeline of hotel search systems, there are three main stages Li, Zhou, Xiao, Huang, Chen, Chen and Xian (2022); Cao, Zhou, Huang, Xiao, Chen and Chen (2022): 1) Matching, which uses the input check-in location and dates to retrieve candidate hotels from a large hotel database; 2) Ranking, which quickly sorts these candidate hotels

[⋆] This work was supported by National Key Research and Development Program of China(2020YFB1710004), and NSFC grant No. U1711261.
[*]Corresponding author
madenghao5@gmail.com (D. Ma); peihongbin@xjtu.edu.cn (H. Pei); chenyueguo@ruc.edu.cn (Y. Chen); baopeng@bjtu.edu.cn (P. Bao); shenliang03@meituan.com (L. Shen)
ORCID(s): 0000-0001-7597-9354 (D. Ma)
[1]This is the first author footnote.
[2]https://www.chinatravelnews.com/article/144279

and selects the top thousand as candidates; 3) Reranking, which sorts these thousand candidates by introducing some fine-grained features, such as behavior-based interest preference, and returns the top hundred candidates to users.

Interest modelling aims to model and capture the behavior-based interest preference automatically from user behavior data Pi, Zhou, Zhang, Wang, Ren, Fan, Zhu and Gai (2020). It has been verified to be very important for personalized recommendation systems, and thus it has drawn more and more attention from the communities of academia and industry Pi, Bian, Zhou, Zhu and Gai (2019); Ren, Qin, Fang, Zhang, Zheng, Bian, Zhou, Xu, Yu, Zhu and Gai (2019); Pi et al. (2020). Although the interest modelling has achieved noticeable successes in product recommendations Pi et al. (2019); Ren et al. (2019), it has not been explored in hotel search that is largely different from product recommendation systems. This motivates us to explore this important topic in hotel search. Considering the online computational cost, we study interest modelling in the reranking stage.

We find that existing interest modelling methods in other domains can not be well applied to hotel search because of two key challenges— high behavior sparsity and large interest gap.

Challenge one: high behavior sparsity. Hotel search suffers from highly sparse behavior data. Hotels are not daily necessities for most people, and thus the behaviors of a user on a hotel booking platform are usually very infrequent. For instance, in Meituan-Hotel, only 0.05% of users click more than 1000 hotels in five months (more details in Section 4.2.1). In contrast, in Taobao, the biggest online shopping platform in China, 23% of users click more than 1000 items in five months Pi et al. (2020), i.e., 460 times of Meituan-Hotel. And most users (95.98%) in Meituan-Hotel click less than 50 hotels in five months. The high behavior sparsity is reflected from two points: for one individual, 1) the behavior sequence is usually short; 2) time intervals between behaviors are always long (details in Survey one of Section 4.2.1). These points keep us from extracting rich semantics to characterize a user's preference upon hotels, because too short sequences may not provide sufficient statistics for pattern learning and the long time intervals significantly weaken the associations between behaviors.

Challenge two: large interest gap. Hotel search suffers from large interest gaps between the historical interests and the current interests. A user searches hotels under a specific query context which may include the check-in location, check-in and checkout dates, holiday, weather, and so on. The context influences both user's interests and user's decisions (such as click or reserve). For example, a user may be interested in a hotel with a good landscape view when she is vacationing, while she may shift her interest to a nearby hotel on a rainy day; When the check-in city is Beijing (the capital of China), a user might be interested in two-star hotels, while interested in four-star hotels when the check-in city is Luoyang (a small city of China), because the price of a four-star hotel at Luoyang is similar to that of a two-star hotel at Beijing. The differences between historical contexts and the current query context often result in the differences between the historical interests and the current interests— interest gap. Further, using clicks from historical interests to model the current interests inevitably suffers the interest gap challenge.

Based on the above challenges, this paper attempts to find answers for the following two **research questions**: 1) How to overcome the high behavior sparsity? 2) How to bridge the large interest gaps?

Insight for challenge one: Deep Click. As mentioned above, it is difficult to learn the rich semantic representation for a click from a sparse click sequence, especially when a click is represented as an id. Accordingly, we propose to extend the semantics of a click from multiple perspectives, and then extend the semantics of a sparse click sequence. To achieve this goal, we analyze the interaction process of a user with a hotel search system and show an example of the process in Figure 1. When a user submits a search request, a hotel e-commerce platform shows a ranked list of hotels (exposure hotels) on the hotel list page. Coming into the list page, the user browses the exposure content of every hotel, such as title, type, distance, upvoting score (upvotes), cumulative order number (orders), price, tags, and so on. If the user likes the exposure content of a hotel, she will click the hotel and come into a hotel detail page. On the detail page, the user may need some time (stay time) to read the specific information (details), share, vote, follow or reserve. The stay time and subsequent actions can be considered as feedback signals which reflect the liking degree of a user w.r.t. hotels. For example, if a user shares a hotel with her friends, the user may like this hotel. Besides, a historical click stems from a historical context, e.g., the check-in location, check-in and checkout dates and holiday, the context implies the conditions under which the click occurred and thus reflects the semantics of the click. So we define a click as a deep click with four components: <exposure content, id, feedback, context>. The four components can model and capture the mutli-view semantic information of a click, so that they can provide richer semantics than an id representation. Some studies use some information to enrich the semantics of a click, such as the time interval between a historical click and the current query Pi et al. (2020), category and brand information Li, Liu, Wu, Xu, Zhao, Huang, Kang, Chen, Li and Lee (2019). Besides, the notion of deep clicks has been implicitly exploited in search engines and

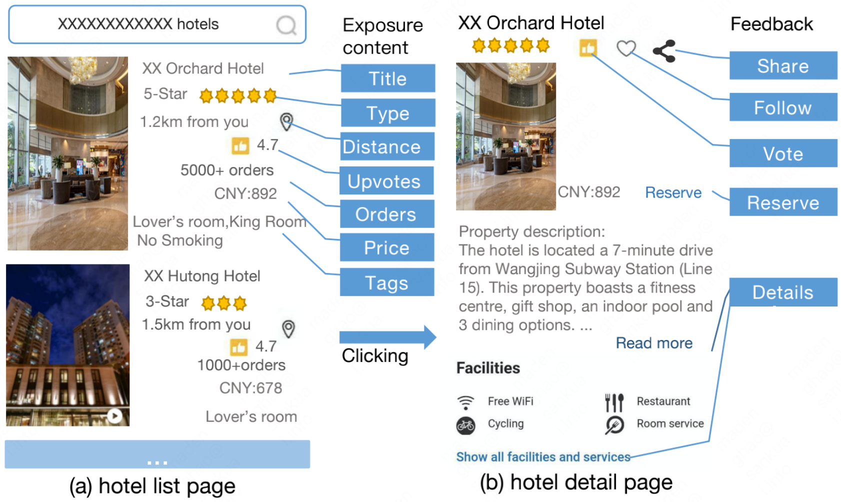


Figure 1: An interaction process of users to hotel search systems: 1) a user submits a query; 2) the system shows a ranked list of hotels with exposure contents; 3) after browsing these contents, the user clicks a hotel; 4) browsing the information of the clicked hotel, the user performs other actions (feedbacks).

in other mining tasks, but we are the first to explicitly and formally define the deep click concept to systematically model the multi-view semantics of a click.

Solution for challenge two. How to unleash the power of deep clicks to bridge the interest gap? An intuitive approach is the concatenating & encoding strategy Li et al. (2019) (C&E), which first concatenates representations of components in a deep click (i.e., exposure content, hotel id, feedback and context), and then encodes the concatenated representations as user's interests. Even C&E strategy has been widely applied to traditional recommendation systems, e.g., product recommendation, it can not take full advantage of deep clicks. This is because users' interests are in the attributes of hotels, e.g., price, hotel-star and distance (exposure content). The context and feedback components in a deep click are not the attributes of a hotel so that they are not suitable for representing the users' interests, although they are suitable for estimating the weights of deep clicks.

Considering the limitation of C&E, we propose the deep clicking interest network (DCIN), which can better exploit deep clicks than C&E. DCIN uses the exposure content and hotel id components to model user's interests on a click, and uses the feedback and context components to estimate the weight of the click. To this end, in DCIN, we propose two cascaded units (Mutual-Attention Interest Calibration Unit and Multi-attention Interest Aggregation Unit) to bridge the interest gap and thus model users' current interests accurately.

Mutual-Attention Interest Calibration Unit (ICU). As motivated in challenge two, a click stems from historical interests which are somewhat different from the current interests, and thus the click can not accurately model the current interests of a user. In other words, the click only represents the user's current interests to some extent. Similarly, a candidate hotel is sorted and selected by the ranking stage and thus represents the current interests to some extent (details in Survey two of Section 4.2.2). According to the two "to some extent", we propose to use the candidate hotel to calibrate the embedding of a click, so that the gap between the calibrated interest embedding and user's current interests can be diminished. To realize this idea, we design the mutual attention unit to learn the weights of a click and a candidate hotel, and then ensure an effective interest calibration.

Multi-attention Interest Aggregation Unit (IAU). The interest calibration unit learns a calibrated interest embedding for every click. How to effectively aggregate these calibrated interest embeddings as the current interest representation? As mentioned above, the context and feedback components are suitable for estimating the weights of clicks. So we argue that 1) if the context of a click is more relevant to the current query context, the calibrated embedding of the click is more important to model user's current interests; 2) if the feedback component of the click includes more positive signals, such as long stay time, sharing and voting actions, the calibrated embedding is more important. Besides, if a calibrated embedding is consistent with the long-term interests of a user, it is important to model user's current interests. Accordingly, we design a multiple attention unit to estimate the weight of a calibrated embedding from the context, user's feedback and interest consistency perspectives, so that the calibrated embeddings, which are relevant to the current query context and important to the user, dominate user's current interests, and thus the interest gap can be further diminished.

We perform offline experiments on a public dataset and a large-scale industrial dataset, and online A/B testing over Meituan-Hotel. The experimental results show that DCIN significantly outperforms the baselines and the improved baselines by using deep clicks. In the online A/B testing, DCIN achieves 2.40% CTCVR (Click Through&Conversion Rate) and 1.09% CTR (Click Through Rate) improvements over the improved baseline models. Note that, DCIN has been deployed with the main traffic of Meituan-Hotel. Our contributions are concluded as follows:

• We are the first to explore the interest modelling task for reranking hotels and identify two key challenges.

• To address the high behavior sparsity challenge, we are the first to formally define the deep click concept to systematically model the multi-view semantics of a click.

• To address the interest gap challenge, the mutual attention interest calibration unit is designed to calibrate the embedding of a click, and the multi-attention interest aggregation unit is developed to estimate the weights of the calibrated embeddings from multiple perspectives.

• We conduct both offline experiments on two large-scale datasets and online A/B testing to validate the effectiveness of the deep clicking interest network.

2. Related Work

2.1. Hotel Search

With the popularity of hotel e-commerce platforms, many studies focus on the hotel search task Zhang, Wang, Wang, Jin and Zhou (2015); Antognini and Faltings (2020); Hsu, Chung and Huang (2003); Liu, Fu, Yao and Xiong (2013). In this section, we make a brief survey about the solutions and datasets for hotel search.

2.1.1. Solutions for Hotel Search

A typical pipeline of hotel search systems often contains three main stages: Matching, Ranking and Reranking, which are introduced in Section 3. Most of all studies focus on the ranking stage. Previous solutions for ranking hotels may be mainly grouped into three classes, i.e., the feature engineering solutions, machine learning solutions and deep learning solutions. In the feature engineering solutions, many features of both hotels and users are designed, such as hotel popularity, hotel star, brand, price, location, ratings, user age, the numbers of clicks, orders and refunds, and so on Adamczak, Deldjoo, Moghaddam, Knees, Leyson and Monreal (2021); Antognini and Faltings (2020). Besides, many work try to construct text-based semantic representations for hotels. In work Ramzan, Bajwa, Jamil, ul Amin, Ramzan, Mirza and Sarwar (2019), the numeric data and textual data are processed by a big data solution by using Hadoop. In work Sharma, Bhatt and Magon (2015), customers' reviews are used for estimating the ratings of hotels by using natural language processing techniques. In work Ambolkar, Bhagat, Buga and Gharat (2022), BERT is used for generating embeddings for hotels from their customers' reviews. These basic features are very important for the machine learning solutions and deep learning solutions. In the machine learning solutions Mavalankar, Gupta, Gandotra and Misra (2019); Zhang et al. (2015); Cozza, Petrocchi and Spognardi (2018), the Boosting Trees, Logistic Regression, SVMRank and Random Forests techniques are developed and improved over many manual features. These solutions aim at finding an effective function to combine these manual features and then rank candidate hotels. In work Kaya (2020), authors use link prediction methods over three different bipartite networks to recommend hotels. In work Ludewig and Jannach (2019), a gradient-boosted learning-to-rank model, Bayesian Personalized Ranking and an embedding model are employed to address the cold-start problem. With the development of the neural networks, many deep learning methods have been proposed to rank candidate hotels. For example, the study Abdool et al. (2020) uses the Recurrent Neural Networks to produce an embedding of the entire query context, and uses the embedding to manage the diversity of hotel results; The study Haldar and et al. (2020) proposes a new ranking neural network to address the positional bias and cold start problems. To the best of our knowledge, few studies focus on the interest modeling in the reranking stage of hotel search.

2.1.2. Datasets for Hotel Search

To support the research of hotel search, some studies and organizations publish the datasets with different focuses. The dataset Antognini and Faltings (2020) emphasizes the customer reviews and voting scores and construct the user's interest preference from the reviews and votes of a user. The datasets [3] emphasize the metadata, such as category, description, facilities, star rating, image count, location and others, which can be used for estimating the quality of a hotel. Recently, the study Adamczak et al. (2021) publishes a session-based hotel recommendations dataset, which emphasizes the interactions of users with hotel e-commerce platforms and is sampled from the trivago website between November 1, 2018, and November 8, 2018. As one of the most popular online travel platforms (OTPs) in China, Alibaba group releases user behaviour data from Fliggy platform [4], which covers all behaviors (including clicks, favorites, adding, and purchases) of approximately 2 million random users from June 3, 2019 to June 3, 2021. It is the largest recommender system dataset from real industrial scenes, and contains user historical behavior data of users and basic attribute data of commodities.

2.2. Interest Modeling for Recommendation System.

Interest modeling for reranking hotels has not been explored. Therefore, we survey the solutions of interest modeling for product recommendation, and compare our solution to them.

2.2.1. Comparison of Click Definitions

In product recommender systems, the click definition can be grouped into two categories. The first category defines a click as a product id Zhou, Zhu, Song, Fan, Zhu, Ma, Yan, Jin, Li and Gai (2018); Zhou, Mou, Fan, Pi, Bian, Zhou, Zhu and Gai (2019); Sun, Liu, Wu, Pei, Lin, Ou and Jiang (2019); Feng, Lv, Shen, Wang, Sun, Zhu and Yang (2019). The second category defines a click as a product id and some side information. The side information is to expand the semantics of a click. In SIM proposed by the study Pi et al. (2020), the time interval between a historical click and the current query is used for extending the semantics of a click. In MIND proposed by study Li et al. (2019), the category and brand information of a product are used for extend the semantics. Since the hotel domain has a higher behavior sparsity than the product domain Hsu et al. (2003); Liu et al. (2013), we need to discover more effective methods to expand the semantics of a click. Therefore, we propose the deep click concept to model the multi-view semantics of a click. A deep click contains four components <exposure content, hotel id, feedback, context>. The exposure contents are the key attributes and contents of a hotel, which can provide richer semantics than hotel id. The feedback can be used for estimating the liking degree of a user to a hotel. The context component is the condition in which a click occurred and it is very important to estimate the weight of a click in hotel domain.

2.2.2. Comparison of Solutions

In traditional recommender systems, such as product recommender systems, the interest modeling has attracted much attention Zhou et al. (2018); Quadrana, Karatzoglou, Hidasi and Cremonesi (2017); Feng et al. (2019). Existing solutions can be divided into two categories: 1) click-independent Zhou et al. (2018); Li et al. (2019), which views historical behaviors as independent units, and applies deep neural network and attention techniques to encode each behavior; 2) click-dependent Quadrana et al. (2017); Zhou et al. (2019); Sun et al. (2019); Feng et al. (2019); Pi et al. (2020), which views users' behaviors as dependent units, and applies long short-term memory Findler (1972), gated recurrent unit Chung, Gülçehre, Cho and Bengio (2015) or transformer Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin (2017) to encode users' interests. Compared to the first category, the second has a higher computation cost since its time complexity is $O(n^2)$ and that of the first category is $O(n)$. According to the different definitions of clicks, two types of encoding approaches are widely applied to industrial applications. The most popular is the concatenating & encoding method (C&E), which firstly concatenates the representations of components in a click, and then encodes each click over its concatenated representation Zhou et al. (2018); Li et al. (2019); Pi et al. (2020). Another type is encoding each component independently Gu, Ding, Wang, Zou, Liu and Yin (2020). For example, the study Gu et al. (2020) parallelly runs the modeling solution on the click, add-to-cart, and payment sequences. As the number of click components increases, its computation cost also increases exponentially.

We propose the deep clicking interest network (DCIN) to better exploit deep clicks than the above encoding approaches. First, DCIN uses the exposure content and hotel id components to model user's interests on a click. Second, the feedback and context components are used for estimating the weight of a click. Besides, to address the large interest gap challenge, we design the mutual attention interest calibration unit which uses the candidate hotel to calibrate the embedding of every click. The multi-attention interest aggregation unit is developed to estimate the

[3] https://data.world/datafiniti/hotelreviews,https://www.kaggle.com/PromptCloudHQ/ hotels-on-goibibo,https://www.kaggle.com/PromptCloudHQ/hotels-on-makemytrip

[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=113649

weight of a calibrated representation from the context, user's feedback and interest consistency perspectives, and then aggregate the weighted representations as the final interest representation.

3. Preliminaries

3.1. Matching

The matching stage is to retrieve candidate hotels from a large hotel database. When using a hotel search system, a user needs to specify a check-in location, check-in and checkout dates. According to the inputs, the matching stage first uses the check-in location to retrieve candidate hotels. Specifically, if the location is a geographical area (such as Beijing, New York and Chicago), all hotels in the area are selected as candidates; If the location is a geographical point (such as Columbia University, New York University and Cornell University), the hotels within 10km from the location are selected as candidates. Secondly, the matching stage checks the status of the retrieved candidate hotels from the check-in date to the checkout date, and filters the hotels with closed status.

3.2. Ranking

The matching stage often retrieves many candidates, and the reranking phrase often uses a complex and effective model to rank the candidate hotels. If too many candidates are directly inputted into the reranking stage, the computation cost of the whole search pipeline will increase dramatically. To address the computation cost problem, the ranking stage is designed to quickly select the top thousand candidate hotels from retrieved candidates, and transfer these thousand candidates to the reranking stage. Therefore, the model in this stage is simpler and more efficient than that in the reranking phrase. Different hotel search systems may have different solutions in the ranking stage. For example, in the e-commerce platform of Meituan-Hotel, the XGBoost Chen and Guestrin (2016) over many statistical features is applied to the ranking stage.

3.3. Reranking

The hotel business is concerned with both Click-Through-Rate (CTR) and Click Through&Conversion Rate (CTCVR). In a search q, for a hotel d, a user u may click, reserve, or not click it. The CTCVR prediction is to estimate the probability that u will click and reserve d, denoted as $\hat{y}_o$ Ghose and Yang (2008). The CTR prediction is to estimate the probability that u will click d, denoted as $\hat{y}_c$. According to the CTCVR or CTR probability, the candidates can be ranked in descending order and finally presented to users. We formulate the probability estimations as follows:

$$\hat{y}_c = p_c(d|u,q) = f_c(\overline{u}, \overline{d}, q, \Psi); \tag{1}$$

$$\hat{y}_o = p_c(d|u,q) \cdot p_o(d|u,q) = f_c(\overline{u}, \overline{d}, q, \Psi) \cdot f_o(\overline{u}, \overline{d}, q, \Psi) \tag{2}$$

where $p_c(d|u,q)$ is the probability that d is clicked by the user u under query q. The $p_o(d|u,q)$ is the probability that d is reserved by the user u under query q, after clicking. Therefore, $p_c(d|u,q) \cdot p_o(d|u,q)$ can be used for estimating the Click Through & Conversion Rate (CTCVR). The $\overline{d}$ and $\overline{u}$ are the features of d and u, respectively. The $f_c()$ and $f_o()$ with parameters Ψ are functions which are to estimate the probability $\hat{y}_c$ and $\hat{y}_o$ by inputting u, d and q.

Features. The user's feature $\overline{u} = (\hat{u}, R(u))$ includes profiling feature $\hat{u}$ and a sequential feature $R(u)$. The user profiling feature includes gender, age level, average consume price level, resident city and so on. The hotel's feature $\overline{d} = (\hat{d}, E(d))$ contains implicit feature $\hat{d}$ and explicit feature $E(d)$. The $\hat{d}$ contains CTR, CVR, and CTCVR. The $E(d)$ includes the hotel type, star, tags, cumulative order number, voting number, the lowest price and location. In this paper, we focus on constructing the feature $R(u)$.

4. Deep Clicking Interest Network

We study the interest modelling task in the reranking stage of a hotel search system, i.e., construct the feature $R(u)$. In this section, we first formally define the interest modelling task for reranking hotels and then present our solution, i.e., deep clicking interest network. Notations used in this paper are shown in Table 1.

Table 1
Notations and Descriptions.

Notation	Description.
u,d	One user and one candidate hotel.
V_u, v_i	Click sequence of u and the i-th click.
$e(v_i)$, $e(d)$	Embeddings of v_i and d
$\overline{u}$, $\overline{d}$	Features of user u and hotel v
$R(u)$	Feature of u learned from click sequences V_u
id_i,ec_i	The embeddings of id and exposed content of v_i
fb_i, c_i	The embeddings of feedback and context of v_i
c_q	The context embedding of the current query q
S	Training samples.
y_o,y_c,$\hat{y}_o$,$\hat{y}_c$	Ground truth labels and predicted CTCVR, CTR
ϕ_*	Activation function.
W_*, b_*	Weight matrix and bias matrix
$\cdot$,$\oplus$	Multiply and concat

4.1. Problem Definition

As motivated in Section 1, the users' behavior sequences in the hotel domain are very sparse, and user's current interests are influenced by the current query context. Since the interest modelling task that we study is in the reranking stage which follows the ranking stage, it can take the candidate hotels selected by the ranking stage as input. So we define the interest modelling task in the reranking stage of hotel search as follows: *The input consists of a user u, a sparse click sequence V_u of u, a candidate hotel d and the current query context c_q. The output is an embedding $R(u)$ that represents the user's current interests.*

In our experiments, the query context includes the check-in location, check-in and checkout dates, holiday, weekend, workday, month, season, quarter and weather. We formulate the interest modelling task as follows:

$$R(u) = f_r(V_u, d, c_q, \psi_r). \tag{3}$$

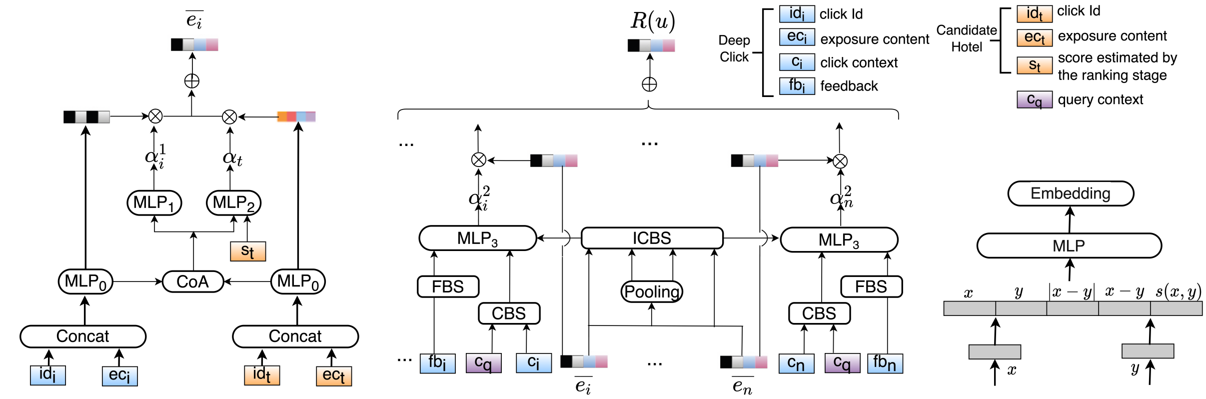
where ψ_r is set of parameters and $f_r()$ is a function which takes V_u, d, c_q and ψ_r as input, and outputs an interest representation $R(u)$. In this paper, we attempt to design an effective function $f_r()$ with parameters ψ_r.

4.2. Solution

We introduce the deep clicking interest network (DCIN) to model user's interest representation $R(u)$. Specifically, we propose the deep click concept to model the multi-view semantics of a click and then enrich the semantics of both a click and a click sequence. So deep clicks can address the problem caused by the behavior sparsity , i.e., high behavior sparsity keeps us from extracting rich semantics to characterize a user's interest preference. Based on deep clicks, two cascaded units are designed to address the interest gap challenge (see Figure 2). The mutual-attention interest calibration unit uses the candidate hotel to calibrate the embedding of a click, so that the gap between the calibrated embedding and user's current interests can be diminished. The multi-attention interest aggregation unit learns the weight of calibrated embedding of a click from the context, user's feedback and interest consistency perspectives, so that the calibrated embeddings, which are relevant to the current query context and important to the user, can dominate user's current interests, and then the interest gap can be further diminished.

4.2.1. Deep Clicking Insight

Survey one: high behavior sparsity. As motivated in Section 1, in the hotel domain, users' click behaviors are very sparse. To verify this, we conduct a survey over a public dataset (Fliggy) and an industrial dataset (Meituan-Hotel). Specifically, we randomly sample 10000 users from the two datasets, respectively. Their click behaviors are extracted and then analyzed. The results are presented in Figure 3. Figure 3 (a) and (b) show the proportion of users with different click numbers in five months (2021-01-01 to 2021-05-31 of Fliggy, and 2021-03-08 to 2021-08-08 of Meituan-Hotel). E.g., $[0, 10)$, 8.98% in (a) means that 8.98% users click less 10 hotels. It can be seen that 1) in Meituan-Hotel, 86.24% of users click less than 20 hotels, 95.98% of users click less than 50 hotels and only 0.05% of users click more than 1000 hotels; 2) In Fliggy, 30.42% of users click less than 20 hotels, 67.93% of users click less than 50 hotels and only



(a) Mutual–attention Interest Calibration Unit (b) Multi–attention Interest Aggregation Unit (c) CoA operation

Figure 2: Deep Clicking Interest Network. (a) and (b) shows the architectures of mutual-attention interest calibration unit and multi-attention interest aggregation unit. The aggregation unit uses three subnets to estimate the weights of calibrated embeddings from the calibration unit. MLP_* denotes a multi-layer perceptron neural network, FBS is a feedback-based subnet, CBS is a context-based subnet and ICBS is an interest consistency based subnet.

0.14% of users click more than 1000 hotels. Figure 3 (c) and (d) show the proportion of the average date intervals between clicks. E.g., "[30, 30+), 74.99%" in (c) means that the date intervals of 74.99% clicks are larger than 30. We can see that 60.6% of date intervals are larger than 30 days in Meituan-hotel, and 74.99% of date intervals are larger than 30 days in Fliggy dataset. This survey illustrates that in the hotel domain, 1) the users' click sequences are usually short; 2) the date intervals among clicks are always long.

Deep Clicks. The high click sparsity keeps us from extracting rich semantics to characterize a user's preference, because the short click sequence may not provide statistical semantic patterns, and the long date intervals weaken the dependent relationships among clicks. To extract rich semantics, an intuitive approach is to extend the semantics of each click in a click sequence. Some studies about product recommendation enrich the semantics of a click from a single perspective. For example, the work Pi et al. (2020) uses the time interval between the time of a historical click and the current query to extend the semantics of a click; The work Li et al. (2019) uses category and brand information to extend a click. Different from these studies, we propose to construct the multi-view semantics for a click and define a click as a structure (deep click) with four components <exposure content, hotel id, feedback, context>.

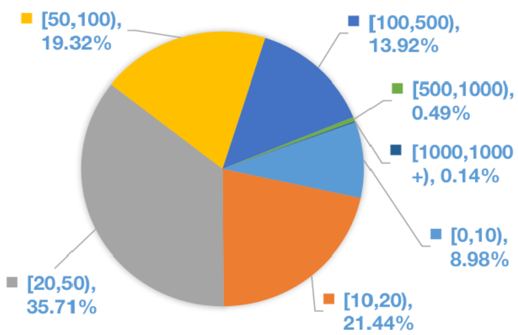
•*Exposure Content.* When a user submits a query request, she will be shown a ranked list of hotels (exposure hotels) at the hotel list page. To help consumers quickly understand hotels, many systems present some key information of each exposure hotel to consumers. Take Meituan-Hotel as an example, the exposure content includes the hotel title, distance between a user and the hotel, price, hotel type, tags, votes and cumulative order number (see Fig. 1). If a user does like the exposure content of a hotel, she will click the hotel. The exposure content is a part of information of the hotel, but it highlights the approved information by users. The exposure content explicitly reflects user's interests on a clicked hotel.

•*Click Id.* In the database of hotels, each hotel is assigned with a unique id. So each click corresponds to a hotel id. The implicit interest representation of a user can be learned from the id sequence.

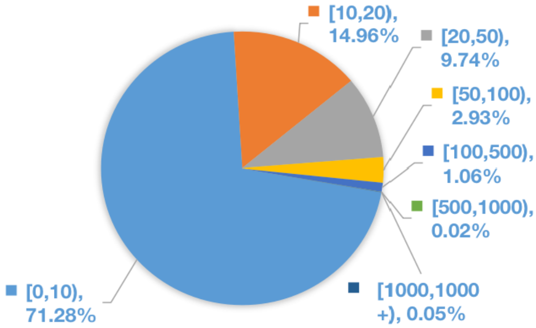
•*Feedback.* After clicking a hotel and coming into the detail page of the hotel, a user may need some time to read the specific information (details), share, vote, follow and reserve. Meanwhile, the subsequent behaviors generate another implicit feature, i.e., stay time. We name the subsequent behaviors and the stay time as feedback signals. The feedback signals can provide clues to estimate the liking degree of the user to the hotel. For example, the longer is the stay time, the more the user likes the hotel; If the user reserved or shared the hotel, the user may like the hotel.

•*Click Context.* The click context is the condition where the click occurred, i.e., the context of the corresponding historical query. So it has the same elements as the current query context. Both the click context and query context contain the check-in location, check-in and checkout dates, holiday, weekend, month, season, quarter and weather.

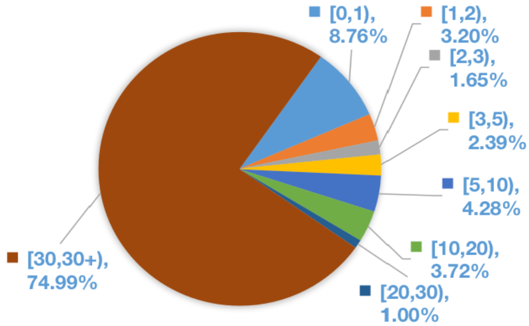
The four components can extend the semantics of a click from different perspectives. Both exposure content and click id can be used for constructing user's interest representation on a click. The feedback and context components can be used for estimating the weights of the interest representations, for well aggregating them as user's current interests.



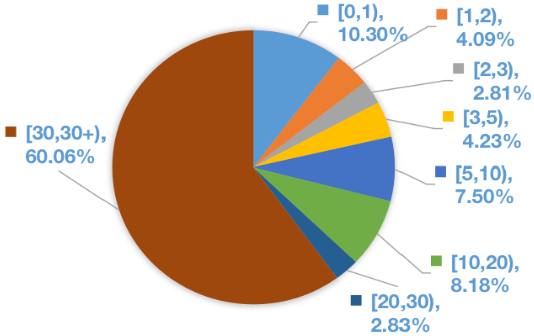
(a) Fliggy: the proportion of users with different numbers of clicks.



(b) Meituan-Hotel: the proportion of users with different numbers of clicks.



(c) Fliggy: the proportion of average date intervals among clicks in the same click sequences.



(d) Meituan: the proportion of average date intervals among clicks in the same click sequences.

Figure 3: The survey about the click sparsity. E.g., "$[0, 10), 8.98\%$" in (a) means that 8.98% users click less 10 hotels, and "$[30, 30+), 74.99\%$" in (c) means that the date intervals of 74.99% clicks are larger than 30. These data illustrate that in hotel domain, users' click sequences are usually short and date intervals among clicks are always long.

Embedding Initialization. In Table 2, we show the initialization of features and embedding. To improve the robustness of models, we need to transform the continuous features to the discrete values, such as distance, price, upvote number, cumulative order number and stay time. Specifically, we map a continuous feature into multiple buckets, use the bucket number to replace the continuous feature value, and then use one-hot vectors to represent them. For the hotel types and tags, we first transform them to the corresponding ids, and then use multi-hot vectors to represent them. For a feature of the one-hot type, we initialize its embedded representation as a single embedding vector with the corresponding dimensions. For a feature of the multi-hot type, we first initialize it as a list of embedding vectors with the corresponding dimensions and then average the list of embedding vectors as the final embedded representation of the feature.

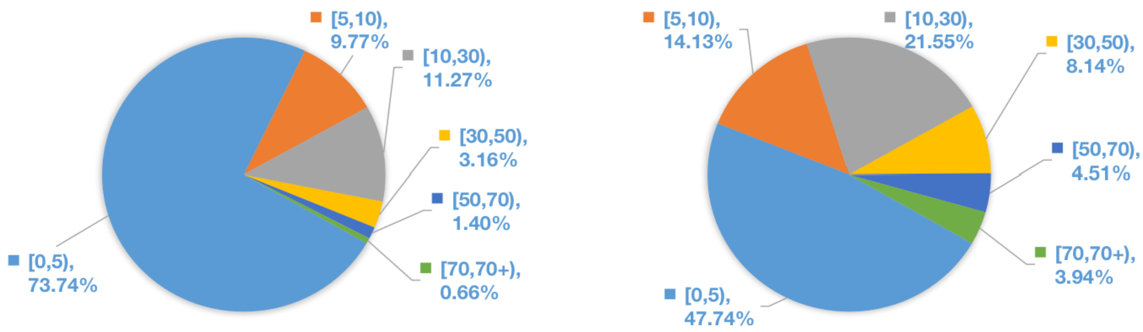
4.2.2. Mutual-Attention Interest Calibration Unit

A historical click stems from a historical context and a historical interests. Since the historical interests may be different from the current interests, the historical click may not completely cover the current interests. For example, a user has three historical clicks corresponding to four-star hotels, and their check-in cities are Luoyang, a small city of China. Now, the user will visit to Beijing, the capital city of China, the interests of the user in star type may be not four-star, because the prices of four-star hotels in Beijing are much higher than these in Luoyang. Therefore, the historical clicks only can represent the current interests to some extent. We study the interest modelling task in the reranking stage. In the reranking stage, the candidate hotels are sorted and selected by the ranking stage. So the candidate hotels are highly relevant to user's current needs, and can also reflect user's current interests to some extent.

Survey two. To verify this point that the candidate hotels selected by the ranking stage can reflect user's current interests to some extent, we make a survey over Meituan-Hotel. Specifically, we investigate the associations between the hotels clicked and reserved by users and their ranks in the ranking stage. The investigated results are presented in

Table 2
Feature and embedding initialization.

Feature	Feature Dimension	Feature Type	Embedding Dimension
Exposure content feature			
id	10^9	one-hot	128
category id	20	one-hot	64
brand id	10^4	multi-hot	128
distance	10^6	one-hot	128
price	10^4	one-hot	128
up-vote number	10^4	one-hot	128
cumulative order number	10^6	one-hot	128
Feedback feature			
sharing feedback	3	one-hot	8
up-voting feedback	3	one-hot	8
following feedback	3	one-hot	8
reserving feedback	3	one-hot	8
stay time	10^6	one-hot	128
Context feature			
check-in city	10^5	one-hot	128
check-in / checkout dates	10^8	one-hot	128
weekend	3	one-hot	8
weather	10^3	one-hot	128
holiday	10^3	one-hot	128



(a) reservations. (b) clicks.

Figure 4: The associations between hotels clicked or reserved by users and their ranks in the ranking stage. E.g., "$[0, 5), 73.74\%$" in (a) means that 73.74% reserved hotels are ranked at top 5 in the ranking stage, and "$[0, 5), 47.74\%$" in (b) means that 47.74% clicked hotels are ranked at top 5 in the ranking stage. These data illustrate that the candidate hotels selected by the ranking stage can represent users' current interests to some extent.

the Figure 4. It can be seen that 1) the ranks of 65.23% reservations are smaller than 10, the ranks of 81.43%reservations are smaller than 30, and the ranks of 93.25% reservations are smaller than 50; 2) the ranks of 53.42% clicks are smaller than 10, the ranks of 67.39% clicks are smaller than 30, and the ranks of 75.61% clicks are smaller than 50. These results illustrate that 1) users are more likely to click and reserve the candidate hotels with lower ranks; 2) the candidate hotels with lower ranks can represent the user's current interests more effectively. The survey verifies that the candidate hotels selected by the ranking stage can be used for modeling the user's current interests.

Since both the historical click and the candidate hotel can represent user's current interests to some extent, we propose to use the candidate hotel to calibrate the embedding of a click, so that the calibrated embedding can well represent user's current interests. To realize this idea, we develop the mutual-attention interest calibration unit and show the unit in Figure 2 (a). The unit first constructs embeddings for both the click and candidate hotel, learns weights for them, and then combine the two weighted embeddings as the calibrated interest representation of the click. The unit is formulated as follows:

$$\overline{e(v_i)} = \alpha_i^1 e(v_i) + \alpha_t e(d), \quad \alpha_i^1 + \alpha_t = 1 \;\; and \;\; \alpha_i^1 > \alpha_t \;\; and \;\; \alpha_i^1 \in (0,1) \;\; and \;\; \alpha_t \in (0,1). \tag{4}$$

where $e(v_i) \in R^N$ and $e(d) \in R^N$ are the embeddings of v_i and d, and N is the dimensionality of $e(v_i)$ and $e(d)$. The scope of each dimension is defined as $[-1,1]$, and thus the elements in $\overline{e(v_i)}$ should be in $[-1,1]$. So we use the constraint $\alpha_i^1 + \alpha_t = 1$ to achieve this. Because the candidate d is used for calibrating the embedding of the click v_i, d can not dominate the semantic of v_i and thus we set $\alpha_i^1 > \alpha_t$. As defined in deep clicks, the exposure content and hotel id can be used for constructing the interest representation. Therefore, we first concatenate the embeddings for the exposure contents and hotel id, and then use a MLP layer to further encode them, i.e., $e(v_i) = MLP_0(ec_i \oplus id_i)$ where ec_i is the concatenated embedding of all exposure contents and id_i is the id embedding of v_i. We use the same way to construct the embedding $e(d)$. The α_i^1 and α_t are the weights of $e(v_i)$ and $e(d)$, respectively. They are learned by the mutual attention between $e(v_i)$ and $e(d)$.

For the estimation of the weight α_t, there are two intuitions: 1) If d is assigned a higher score in the ranking stage, d is more relevant to the user's current needs and represent the user's current interests more accurately. So α_t should be larger. This intuition has been verified in the survey two— users are more likely to click and reserve the candidate hotels with lower ranks; 2) If $e(d)$ is more relevant to $e(v_i)$, the user is more likely to click d and thus d is more important to construct the user's interests. So α_t should be assigned a larger value. Based on the above two intuitions, we estimate the weight α_t as follows:

$$\alpha_t = MLP_1(w_1[e(v_i) \oplus e(d) \oplus CoA(e(v_i), e(v_d)) \oplus s(d)] + b_1) \tag{5}$$

where MLP_1 with parameters w_1 and b_1 is a neural network with $n \times n \times 1$ multi-layer perceptron (MLP), and its activation functions of the first two layers are Relu [5], and that of the third layer is Sigmoid [6] for ensuring $\alpha_i^1 \in (0,1)$ and $\alpha_i^1 \in (0,1)$. The $\oplus$ denotes the concatenating operation. The $s(d)$ is to model the intuition one. We formulate it as $score(d) \oplus nscore(d) \oplus position(d)$. The $score(d)$ is the score of d which is estimated by the ranking stage, and $nscore(d)$ is the normalized score by using Min-max feature scaling method [7]. The $position(d)$ is the rank of d in the ranked list of candidate hotels. The $CoA(e(v_i), e(d))$ is the relevance between $e(v_i)$ and $e(d)$, to capture and model the intuition two. We formulate it as follows:

$$CoA(e(v_i), e(d)) = MLP_r(w_r[e(v_i) \oplus e(d) \oplus |(e(v_i) - e(d))| \oplus (e(v_i) \circ e(d))] + b_r) \tag{6}$$

where MLP_r with parameters w_r and b_r is a $n \times n \times n$ multi-layer perceptron (MLP) neural network and its activation functions of the three layers are Relu. The notation $-$ is the element-wise subtraction, $|\cdot|$ is a sign of computing the absolute value, and $\circ$ is the Hadamard product [8], i.e., the element-wise product.

Like the above second intuition, if $e(v_i)$ is more relevant to $e(d)$, $e(v_i)$ is more likely to be the user's current interests and thus α_i^1 should be larger. We estimate α_i^1 as follows:

$$\alpha_i^1 = MLP_2(w_2 CoA(e(v_i), e(d) + b_2) \tag{7}$$

where MLP_2 is a $n \times n \times 1$ MLP that has the same activation functions of MLP_1 in Equation 5. To ensure $\alpha_i^1 + \alpha_t = 1$ and $\alpha_i^1 > \alpha_t$, we adjust α_i^1 and α_t as follows:

$$\begin{cases} \alpha_{sum} = \alpha_i^1 + \alpha_t; \\ \alpha_i^1 = \alpha_i^1 / \alpha_{sum}, \quad \alpha_t = \alpha_t / \alpha_{sum}; \\ \alpha_i^1 = max(\alpha_i^1, \alpha_{sum}/2), \quad \alpha_t = min(\alpha_t, \alpha_{sum}/2). \end{cases} \tag{8}$$

where $max(\cdot,\cdot)$ returns the largest one from two values and $min(\cdot,\cdot)$ returns the smallest one from two values.

[5] https://en.wikipedia.org/wiki/Rectifier_(neural_networks)
[6] https://en.wikipedia.org/wiki/Sigmoid_function
[7] https://en.wikipedia.org/wiki/Normalization_(statistics)
[8] https://en.wikipedia.org/wiki/Hadamard_product_(matrices)

4.3. Multi-attention Interest Aggregation Unit

The mutual-attention interest calibration unit generates a calibrated interest embedding for every click. The multi-attention interest aggregation unit is to aggregate these calibrated interest embeddings as the final interest representation. In the aggregation, the calibrated embeddings, which are relevant to the current query context and important to the user, are hoped to dominate the user's current interests. We formulate the aggregation as follows:

$$R(u) = \sum_{i=0}^{|V_u|} \alpha_i^2 \ \overline{e(v_i)}, \alpha_i^2 \in (0,1). \tag{9}$$

where $\overline{e(v_i)}$ is the calibrated embedding that learned from the click v_i by using the mutual-attention interest calibration unit, and α_i^2 is the weight of $\overline{e(v_i)}$.

To achieve the effective aggregation, we need to accurately estimate the weight α_i^2. Accordingly, we propose three insights from query context and user perspectives to estimate α_i^2. First, as motivated in Section 1, the context influences both users' interests and user's decisions (such as click or reserve). If the context of a click is more relevant to the current query context, the calibrated embedding of the click (i.e., $\overline{e(v_i)}$) is more important to represent user's current interests and thus $\overline{e(v_i)}$ should be assigned a higher weight. For example, a user has three historical clicks v_1, v_2 and v_3 with check-in cities (one element of the context component in a deep click), Beijing, New York and Chicago. Now, the user plans to visit New York (one element of the current query context) and wants to reserve a hotel in New York. The click v_2 should be assigned a higher weight than other two clicks, because the check-in city of v_2 is the same as the current check-in destination. Second, if the calibrated embedding of a click v_i is consistent with the long-term and stable interests of a user, it is more important to represent user's current interests. Third, if the feedback component of a click v_i contains more positive signals, the user is more likely to like the hotel of v_i and thus $\overline{e(v_i)}$ is more important to represent user's current interests. For example, after clicked the hotel, the user up-voted the hotel. The up-voted feedback is a very positive signal that the user likes the hotel.

Based on the above three insights, we propose the multi-attention interest aggregation unit (IAU) to estimate the weights for the calibrated interest embeddings. We show the architecture of IAU in Figure 2 (b). The architecture of IAU contains four components. The first component, i.e., context-based subnet, aims to estimate the relevance between the context of a click and the current query context, for capturing the insight one. In the second component, i.e., the interest consistency based subnet, the calibrated interest embeddings are first aggregated to model the long-term and stable interests of a user, and then the relevance between a calibrated interest embedding and the aggregated embedding is estimated, for capturing the insight two. The third component, i.e., feedback-based subnet, uses the feedback signals to estimate the weights of clicks, for capturing the insight three. In the fourth component, a MLP neural network takes the encoded results of the previous three components as input, and outputs a weight, i.e. α_i^2. We formulate the estimation of α_i^2 as follows:

$$\alpha_i^2 = MLP_3(w_3[CBS_i \oplus ICBS_i \oplus FBS_i] + b_3) \tag{10}$$

where MLP_3 with parameters w_3 and b_3 is a MLP neural network with $n \times n \times 1$ layers, and its activation functions of the first two layers are Relu, and that of the third layer is Sigmoid for ensuring $\alpha_i^2 \in (0,1)$. The CBS_i is the context-based subnet, $ICBS_i$ is the interest consistency based subnet, and FBS_i is the feedback-based subnet. The subnet CBS_i is formulated as $CoA(c_i, c_q)$, where c_i is the context of v_i and c_q is the current query context. The CoA is to further encode c_i and c_q, and generate their matching representation, which is formulated in Equation 6. We formulate the subnet $ICBS_i$ as $CoA(\overline{e(v_i)}, \widetilde{V_u}) \oplus CoA(\overline{e(v_i)}, e(d))$, where $\widetilde{V_u}$ is the average pooling of click sequence V_u for modelling the long-term and stable interests. Besides, the subset FBS_i is formulated as $s_i \oplus f_i \oplus v_i \oplus re_i$. The s_i, f_i, v_i and re_i are assigned binary values and represent whether the hotel corresponding to v_i has been shared, followed, up-voted, reserved or not.

4.4. Comparisons with Existing Solutions

Comparison with DIN. The deep interest network (DIN) Zhou et al. (2018) is a widely applied solution of modelling user's diverse interests from rich historical behaviors. It uses the candidate to estimate the weights of historical clicks and then aggregates the weighted embeddings as user's current interests. In other words, DIN doesn't extend the semantics of a click. The formulation of DIN is presented as follows:

$$R(u) = \sum_{i=1}^{|V_u|} a(e(v_i), e(d))e(v_i) \tag{11}$$

where $a(\cdot)$ is a feed-forward network to estimate the relevance between $e(v_i)$ and $e(d)$, i.e. weight of $e(v_i)$. Comparing the mutual-attention interest calibration unit (ICU) to DIN (i.e., Equation 4 vs. Equation 11), if α_t in Equation 4 equals 0, ICU is similar to DIN. Just Because $\alpha_t! = 0$ and $e(d)$ represents the current interests to some extent, ICU can use the candidate embedding $e(d)$ to calibrate the semantics of click v_i and then bridge the gap between $e(v_i)$ and user's current interests. Besides, DIN does not use the candidate to calibrate the embedding of a click, but uses the candidate to estimate the weight. Comparing the multi-attention interest aggregation unit (IAU) to DIN (i.e., Equation 9 vs. Equation 11), IAU estimates the weight of a click from multiple perspectives, i.e., context, feedback and interest consistency (see Equation 10), while DIN only uses the candidate to estimate the weight.

Comparison with Transformer-based solutions. Transformer-based models, such as SASRec Kang and McAuley (2018), BST Chen, Zhao, Li, Huang and Ou (2019) and BERT4Rec Sun et al. (2019), use the multi-head self-attention blocks to learn the sequential representations or model users' interests from clicks sequences. Take BST Chen et al. (2019) as an example, we analyze the association between the transformer-based solutions and our proposed mutual-attention interest calibration unit. Given a user's click sequence $V_u = \{v_1, v_2, \cdots, v_i, , \cdots v_n\}$ and a candidate d, BST first concatenates V_u and d as a whole sequence $L = \{v_1, v_2, \cdots, v_n, d\}$. Secondly, BST uses some self-attention blocks to encode L and generates an embedding for each unit v_i. The self-attention component is presented as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}}V) \tag{12}$$

where $Q = LW^Q$, $K = LW^K$ and $V = LW^V$. For a click v_i, we can simplify the self-attention component as follows:

$$\overline{e(v_i)} = \sum_{k=1}^{|V_u|} a(e(v_i), e(v_k))e(v_k) \tag{13}$$

where $a(e(v_i), e(v_k))$ is the relevance between $e(v_i)$ and $e(v_k)$. According to the above inference, BST uses both other clicks and the candidate to extend the semantics of the click v_i. In principle, BST also uses the candidate to bridge the gap between $e(v_i)$ and the user's current interests.

4.5. Training and Inference

According to the current interest representation $R(u)$ constructed by using Equation 9, we rewrite the estimations of CTR and CTCVR probabilities in Equation 1 and 2 as follows:

$$\hat{y_c} = p_c(d|u, q) = f_c(\overline{u}, \overline{d}, q, \Psi) = f_c(\hat{u}, R(u), \hat{d}, e(d), \Psi) = MLP_4(w_4[\hat{u} \oplus R(u) \oplus \hat{d} \oplus e(d)] + b_4) \tag{14}$$

$$\hat{y_o} = p_c(d|u, q) \cdot p_o(d|u, q) = f_c(\overline{u}, \overline{d}, q, \Psi) \cdot f_o(\overline{u}, \overline{d}, q, \Psi) = f_c(\hat{u}, R(u), \hat{d}, e(d), \Psi) \cdot f_o(\hat{u}, R(u), \hat{d}, e(d), \Psi) \tag{15}$$

$$f_o(\hat{u}, R(u), \hat{d}, e(d), \Psi) = MLP_5(w_5[\hat{u} \oplus R(u) \oplus \hat{d} \oplus e(d)] + b_5) \tag{16}$$

where MLP_4 with parameters w_4 and b_4 and MLP_5 with parameters w_5 and b_5 are MLP neural networks with $n \times n \times 1$ layers, and the activation functions of the first two layers are Relu, and that of the third layer is Sigmoid. We use the MMOE technique Ma, Zhao, Yi, Chen, Hong and Chi (2018) to optimize CTR and CTCVR prediction tasks simultaneously. MMoE adapts the Mixture-of-Experts structure to multi-task learning by sharing the expert submodels across all tasks, and uses a gating network to optimize each task. The cross-entropy loss function Zhang and Sabuncu (2018) is applied to estimate the loss of both CTR and CTCVR prediction tasks, and formulated as follows:

$$L_{ctr} = -\frac{1}{|S|} \sum_{<u,d,q,y_c> \in S} \{y_c \log \hat{y_c} + (1 - y_c) \log(1 - \hat{y_c})\} \tag{17}$$

$$L_{ctcvr} = -\frac{1}{|S|} \sum_{<u,d,q,y_o> \in S} \{y_o \log \hat{y}_o + (1 - y_o) \log(1 - \hat{y}_o)\} \tag{18}$$

where S is a set of training data, and $|S|$ is the size of S. The y_c and y_o are the ground truth labels about $< u, d, q >$. The final loss function is the combination of L_{ctr} and L_{ctcvr}, i.e., $L_{ctr} + L_{ctcvr}$. Besides, in the prediction stage, candidate hotels can be sorted based on their CTR or CTCVR probabilities. In our experiments, the CTCVR probability is used for sorting the candidates.

Training data. In a search q, given a candidate hotel $d \in D$, a user $u \in U$ may click d, reserve or not. We can use a tuple to record the response of u on d, i.e., (u, d, q, y_c, y_o) where $y_c = 1$ if u clicks d, otherwise $y_c = 0$; $y_o = 1$ if u reserves d, otherwise $y_o = 0$. From online systems, we can aggregate a lot of tuples as training data.

5. Experiments

5.1. Experiment Setup

Overview of Objectives. We first review the research questions introduced in Section 1, i.e., How to overcome the high behavior sparsity? How to bridge the large interest gaps? With respect to the two research questions, we design several experimental objectives and then verify the effectiveness of our model based on those objectives. The experimental objectives are presented as follows:
•O1: Can DCIN better address the high behavior sparsity and large interest gap challenges than baseline models?
•O2: Can DCIN exploit the deep clicks more effectively than the concatenating & encoding strategy (C&E)?
•O3: Ablation Study. What is the effectiveness of components in DCIN?
•O4: Can DCIN achieve better business metrics than the improved baselines?
•O5: What is the computational cost of different models?

Comparison Solutions. As depicted in Section 2, the interest modelling for reranking hotels has not been explored. So we select some state-of-the-art solutions from product recommendation as our baseline models. In product recommendation, the existing interest modelling solutions can be divided into two categories: click-independent and click-dependent. We select some novel solutions from the two categories as baselines. Besides, to fairly compare the effectiveness of all models, we improve these state-of-the-art solutions by using the deep clicks and query context. In experiments, we compare our model to both the baseline models and improved baseline models. The baseline models are presented as follows:
• Base. The model only takes the user and hotel profile features as input, and any interest modelling solutions are not applied into this model.
• DNN. It uses a three-layer MLP to encode all clicks, and averages their representations as user's interests.
• DIN Zhou et al. (2018). It uses the candidate to estimate the weights of clicks and aggregates the weighted representations as the final interest representation of a user.
• DIEN Zhou et al. (2019). It uses an interest extractor layer to capture temporal interests and interest evolving layer to model interest evolving process.
• BST Sun et al. (2019). It uses the deep bidirectional self-attention model to capture user's interests.
• SIM Pi et al. (2020). It first searches the important clicks, and then uses DIN Zhou et al. (2018) to model user's interests over the important clicks, for reducing the computation cost.

We use our proposed deep clicks and query context to improve all baselines based on the C&E strategy, and denote the improved baselines as DNN++, DIN++, DIEN++, BST++ and SIM++. Specifically, we concatenate the representations of components in a deep click as the representation of the click, concatenate representations of the query context, hotel id and explicit feature of a candidate hotel as the representation of the hotel, and run baselines on the concatenated representations.

Dataset. We perform offline experiments on a public dataset (Fliggy) [9] and a large scale industrial dataset (Meituan-Hotel), and online AB testing on the e-commerce platform of Meituan-Hotel. The Fliggy dataset is provided by Fliggy Trip Platform, one of the most popular online travel platforms (OTPs) in China. It contains all behaviors (including clicks, favorites, adding, and purchases) of approximately 2 million random users and from June 3, 2019 to June 3, 2021. Since the dataset does not contain the unclick behaviors, we can not run the CTR prediction task. Therefore,

[9] https://tianchi.aliyun.com/dataset/dataDetail?dataId=113649

Table 3
Dataset Statistics. "Days" denotes the number of days to which samples belong. "Negative" and "Positive" denote the numbers of negative samples and positive samples.

	Train	Dev	Test
Fliggy			
Days	577	89	63
Negative	4,269,540	3,803,355	5,051,248
Positive	5,758,091	407,709	542,832
Meituan-Hotel			
Days	56	7	7
Negative	2,601,917,923	89,354,176	79,333,178
Positive	243,131,792	9,124,478	8,156,981

we use the historical "purchase" and "'adding' sequences to predict the probabilities of next purchasing and adding. In our experiments, we use the data from June 3, 2019 to December 31, 2020 as training set, that from January 1, 2021 to March 31, 2021 as validation set, that from April 1, 2021 to June 3, 2021 as test set. The historical behaviors of five months are used for modelling users' interests. The Meituan-Hotel data is randomly sampled from the logs in Meituan-Hotel, the biggest online hotel booking platform in China. The dataset contains approximately 7 million random users and 1.5 million hotels. Each user is assigned with a behavior sequence of the last five months. The maximum length of the behavior sequences is 500 and the minimum length is 0. According to the method of sampling Wu, Qiao, Chen, Wu, Qi, Lian, Liu, Xie, Gao, Wu and Zhou (2020), four weeks' samples (from June 1, 2021 to July 26, 2021) are used for training, samples of the following two weeks are used for evaluating and testing, respectively. Note that the training examples is from eights weeks' logs, but the click sequence of every user is from five months' logs. The data statistics of both Fliggy and Meituan-Hotel are shown in Table 3. "Negative" refers to the number of negative samples which are exposed but not clicked. "Positive" refers to the number of positive samples. The specific survey about behavior sequences is introduced in Section 4.2.

Evaluation Metrics. For offline experiments, we use the following metrics to evaluate the performance of models:
• AUC Fawcett (2006). Area under the curve (AUC) is a widely used metric in classification and ranking tasks.
• QAUC. Studies Zhu, Jin, Tan, Pan, Zeng, Li and Gai (2017); Zhou et al. (2018) introduce a user-centric weighted AUC, which measures the goodness by averaging AUC over users.
• QNDCG Wang, Wang, Li, He and Liu (2013). QNDCG is the average normalized discounted cumulative gain (NDCG) of all queries.
• Step/sec. It refers to the executed steps per second, measuring the running efficiency of a model.

For online A/B testing, we use two core metrics: CTR and CTCVR. For the online computational cost, we use the TP99 latency as a metric, which is time consumption of the top 99.99% queries Liu, Lu, Zhao, Xu, Peng, Liu, Zhang, Li, Jin, Bao and Yan (2020).

Reproducibility. We keep hyper parameters of all models as the same default values for fair comparisons. Specifically, batch_size is 32, the optimizer is the Adam with $1e-6$ learning_rate, embed_optimize is the Adagrad optimizer with 0.8 learning_rate, the weights of CTCVR and CTR are 1. All models are developed in Python 2.7 and Tensorflow 1.10. Tensorflow is distributed on 40 computers, and each computer has a 4-core cpu and $10G$ memory. The statistical significance is tested against DCIN by using a two-tailed paired t-test, and † denotes the difference at 0.05 level.

5.2. Experimental Results

5.2.1. Comparison with baselines.

To verify the experimental objective O1, we perform all models over the Meituan-Hotel and Fliggy datasets, and show their metrics in Table 4 and 5. As depicted in Section 3, the interest modelling is one component of the reranking framework. So we can use any one baseline to replace our proposed interest modelling model, i.e., DCIN. The metrics in Table 4 and 5 reflect the effectiveness of the reranking framework with different interest modelling models. From Table 4 and 5, it can be seen that DCIN significantly outperforms all baselines on all metrics. Interestingly, the advantage of DCIN in Table 4 is less significant than that in Table 5. This is because the Meituan-Hotel dataset includes 600+ profile features, while the Fliggy dataset only contains several profile features; These large numbers of profile features have been able to characterize the user's interests and needs, and many models supported by these large numbers of

Table 4
Metric comparison over the Meituan-Hotel dataset. Since AUC is a number, not a list, the significance of AUC metrics is not tested.

Model	AUC	QAUC	QNDCG
Base	0.90148	$0.78335^{\dagger}$	$0.72678^{\dagger}$
DNN	0.90262	$0.78641^{\dagger}$	$0.72992^{\dagger}$
DIN	0.90302	$0.78738^{\dagger}$	$0.73142^{\dagger}$
SIM	0.90313	$0.78853^{\dagger}$	$0.73179^{\dagger}$
DIEN	0.90197	$0.78428^{\dagger}$	$0.72693^{\dagger}$
BST	0.90207	$0.78532^{\dagger}$	$0.72763^{\dagger}$
DCIN	**0.90567**	**0.79023**	**0.73387**

Table 5
Metric comparison over the Fliggy dataset.

Model	AUC	QAUC	QNDCG
DNN	0.71429	$0.60646^{\dagger}$	$0.79936^{\dagger}$
DIN	0.73032	$0.60845^{\dagger}$	$0.79903^{\dagger}$
SIM	0.73107	$0.60803^{\dagger}$	$0.79868^{\dagger}$
DIEN	0.72861	$0.60158^{\dagger}$	$0.79659^{\dagger}$
BST	0.71952	$0.60041^{\dagger}$	$0.79577^{\dagger}$
DCIN	0.73776	0.62053	0.80456

Table 6
Metric comparison over the Meituan-Hotel dataset without profile features.

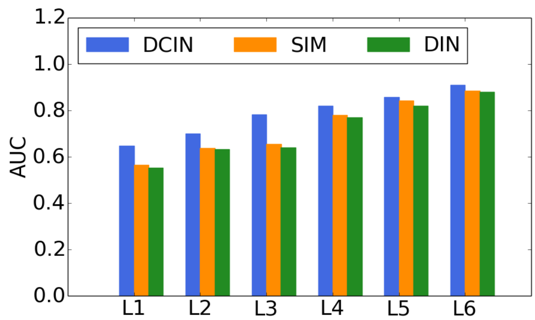
Model	AUC	QAUC	QNDCG
DNN	0.62102	$0.54554^{\dagger}$	$0.54402^{\dagger}$
DIN	0.65302	$0.57876^{\dagger}$	$0.57113^{\dagger}$
SIM	0.66906	$0.59959^{\dagger}$	$0.59512^{\dagger}$
DIEN	0.62540	$0.55120^{\dagger}$	$0.55924^{\dagger}$
BST	0.64534	$0.56931^{\dagger}$	$0.56942^{\dagger}$
DCIN	**0.79757**	**0.66077**	**0.63191**

profile features can achieve high metrics. To verify this point, we run all models over the Meituan-Hotel dataset without profile features, and show the experimental results in Table 6. In Table 6, the advantage of DCIN is more significant than that in Table 4. These significant metric improvements achieved by DCIN in Table 4, 5 and 6 can positively verify the experimental objective O1: DCIN can better address the high behavior sparsity and large interest gap challenges than baseline models.

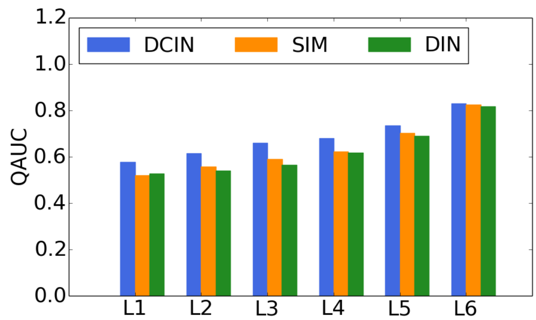
We use the query context and deep clicks to improve the baselines according to the concatenating and encoding strategy (C&E). Specifically, the representations of components in a deep click are concatenated as the representation of the click. The representations of the query context, candidate hotel id and / or hotel profile features are concatenated as the representation of the candidate hotel. The baseline models are performed over the concatenated representations. The improved baseline models are denoted as X++ where X refers to a baseline model. The experimental results of DCIN and the improved baselines are shown in Table 7. We can see that the improved baselines perform better than original baselines. The baseline DIN, SIM, DIEN, BST ignore the characterization of hotel search, i.e., query context. Besides, DIN, DIEN and BST view a click as a hotel id, and SIM views a click as a combination between the hotel id and time interval from a historical click to the current query. In other words, DNN, DIN, SIM, DIEN and BST can not use the query context and deep clicks to guild their interest modelling. But the improved baseline models can use both query context and deep clicks to model user's interests. The metric improvements between the improved baselines and original baselines further verify the experimental objective O1, i.e., deep clicks can provide richer semantics than the hotel ids and the time internals so that the problem caused by the behavior sparsity can be addressed effectively.

Table 7
Using deep clicks to improve baselines on the Meituan-Hotel and Fliggy dataset. The "++" denotes that the baselines are improved by using deep clicks according to C&E strategy.

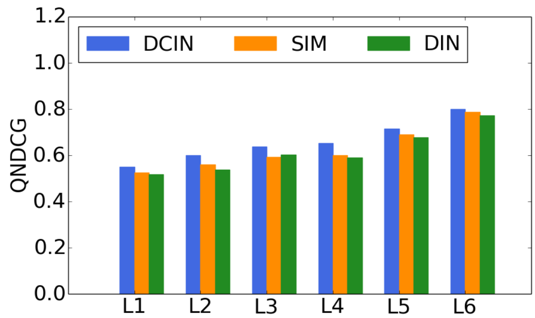
Metric	DIN	DIN++	SIM	SIM++	DIEN	DIEN++	BST	BST++	DCIN
Meituan-Hotel									
AUC	0.90302	0.90405	0.90313	0.90423	0.90197	0.90295	0.90207	0.90327	**0.90567**
QAUC	0.78738	0.78873	0.78853	0.78913	0.78428	0.78473	0.78532	0.78552	**0.79023**
QNDCG	0.73142	0.73223	0.73179	0.73216	0.72693	0.72734	0.72763	0.72813	**0.73387**
Meituan-Hotel without profile features									
AUC	0.65302	0.68561	0.66906	0.71643	0.62540	0.66734	0.64534	0.70356	**0.79757**
QAUC	0.57876	0.59023	0.59959	0.61298	0.55120	0.58276	0.56931	0.58965	**0.66077**
QNDCG	0.57113	0.59901	0.59512	0.60712	0.55924	0.58743	0.56942	0.59804	**0.63191**
Fliggy									
AUC	0.73526	0.74665	0.74107	0.75363	0.72861	0.74381	0.71952	0.72047	
QAUC	0.61131	0.61490	0.61332	0.61361	0.60158	0.61337	0.60041	0.60832	
QNDCG	0.80143	0.80226	0.80126	0.80335	0.79659	0.80119	0.79577	0.79972	



(a) AUC.



(b) QAUC.



(c) QNDCG.

Figure 5: The effect of click sequence lengths on models. L1, L2, L3, L4, L5 and L6 denotes the sequence lengths range from $[0, 10)$,$[10, 30)$,$[30, 50)$,$[50, 100)$,$[100, 150)$ and $[150, 150 + +)$, respectively

From Table 7, it can be seen that DCIN performs better than the improved baseline models. The improved baselines adopt the concatenating & encoding strategy Li et al. (2019) where all components in a deep click are concatenated as the semantic representation of a click. But the feedback and click context components are not the features of hotels, and thus they are not suitable for representing user's interests. Considering this point, DCIN uses the feedback and click context to estimate the weights of clicks, and uses the exposure contents and hotel id to model user's interests. These above metric comparisons verify the experimental objective O2: DCIN can exploit the deep clicks more effectively than the concatenating & encoding strategy (C&E).

Table 8
Effectiveness of components in DCIN. The "'-*' denotes that the "*" component is not applied to DCIN.

Metric	DCIN-ID	DCIN-EC	DCIN-ICU	DCIN-IAU	DCIN-CBS	DCIN-ICBS	DCIN-FBS	DCIN
AUC	0.75536	0.77903	0.73612	0.69473	0.72065	0.78339	0.76653	**0.79757**
QAUC	0.64834	0.65331	0.64503	0.63296	0.64178	0.65781	0.65001	**0.66077**
QNDCG	0.62132	0.62549	0.61975	0.61039	0.62165	0.62834	0.62375	**0.63191**

We investigate the effect of click sequence lengths on the three metrics of DCIN, DIN and SIM. The investigated results are shown in Figure 5. With the increase of sequence lengths, the metrics of three models gradually get better. This is because the long click sequences can provide rich semantics to the interest modeling solutions so that the solutions achieve good performance metrics. Besides, the advantage of DCIN is more significant when the click sequence lengths are shorter. On the short click sequences, the DIN and SIM models can not extract rich semantics and then can not accurately capture the user's interests. But the DCIN uses the deep click concepts to capture the multiple-view semantics of clicks and then models user's interests well.

5.2.2. Ablation Study

To verify the effectiveness of components in DCIN, i.e. the experimental objective O3, we conduct an ablation study over the dataset Meituan-Hotel without the profile features, and show the results in Table 8. The DCIN-EC and DCIN-ID refer to that the exposure contents and hotel id are not applied into DCIN, respectively. Comparing the metrics of DCIN, DCIN-ID and DCIN-EC, we can see that DCIN performs better than DCIN-ID and DCIN-EC. This verifies that the combination of hotel id and exposure contents can better represent user's interests than any single one. The exposure contents contain more elements, i.e., category, brand, distance, price, up-vote number and cumulative order number, and can explicitly represent user's interests. Incorporating the exposure contents can provide rich semantics for a click and well model user's interests. The metric gaps between DCIN and DCIN-EC illustrates that extending the semantics of clicks can diminish the effect of the behavior sparsity.

In Table 8, DCIN-ICU refers to that the mutual attention interest calibrate unit is not applied to DCIN and replaced by the DNN baseline. DCIN-IAU refers to that the multi-attention interest aggregation unit is not applied to DCIN and replaced by the average pooling solution. DCIN-CBS, DCIN-ICBS and DCIN-FBS denote that the context-based subnet, the interest consistency subnet, and the feedback-based subnet are not applied to the multi-attention interest aggregation unit, see Equation 10. It can be seen that 1) DCIN outperforms DCIN-ICU and DCIN-IAU models. The metric gaps between DCIN and DCIN-ICU verify the effectiveness of the mutual attention interest calibration unit. The metric gaps between DCIN and DCIN-IAU verify the effectiveness of the multi-attention interest aggregation unit; 2) DCIN-ICU performs better than DCIN-IAU, which illustrates that the multi-attention interest aggregation unit is more important to construct the user's interests than the mutual-attention interest calibration unit.

The metrics of DCIN-ICBS are better than those of DCIN-FBS, and the metrics of DCIN-FBS are better than those of DCIN-CBS. This illustrates that the context-based subset is more important than other two subnets, and the feedback-based subset is more important than the interest consistency based subnet. In hotel search, user's decisions and interests depend on the current query context, such as a check-in city, check-in and checkout dates. By using the context-based subset, the clicks relevant to the current query context are assigned as high weights, so that they can dominate user's interests. The metric gaps between DCIN and DCIN-CBS verify the effectiveness of the context-based subnet. As motivated in Section 1, the feedback component of a deep click can be used for estimating the liking degree on the hotel corresponding to the click. For example, after clicking a hotel, a user shared the hotel with her/his friends. The sharing feedback can reflect that the user liked the hotel. The feedback-based subnet lets the clicks with more positive feedbacks have higher weights. The metric gaps between DCIN and DCIN-FBS verify the effectiveness of the feecback-based subnet. In the interest consistency based subnet, if a click is consistent with the long-term and stable interests of a user, the click will be assigned a high weight. Comparing the metrics between DCIN and DCIN-ICBS, we conclude that the interest consistency based subnet is important to estimate the weights of clicks. The metric comparisons among DCIN, DCIN-CBS , DCIN-ICBS and DCIN-FBS verify the effectiveness of the context-based subnet, the interest consistency subnet, and the feedback-based subnet in IAU.

Table 9
Online A/B testing (relative improvements). Each model is assigned with 20% traffic.

Model	CTCVR	CTR
DIN++	0.00%†	0.00%†
SIM++	+0.68%†	+0.31%†
DCIN	**+2.40%**	**+1.09%**

Table 10
Comparisons of computational cost.

Metric	DNN++	DIN++	SIM++	DIEN++	BST++	DCIN
Step/Sec	60	55	53	20	22	50
TP99	—	39.4ms	40.0ms	—	—	40.1ms

5.2.3. Online A/B testing.

To verify the experimental objective O4, we perform the A/B testing on the e-commerce platform of Meituan-Hotel and show the results in Table 9. Considering the computational cost and economic benefits, we select DIN++, SIM++ and DCIN to conduct A/B testing for two weeks. Each model is assigned with 20% traffic. It can be seen that DCIN achieves the best online metrics. Compared to DIN++, DICN achieves the +2.40% CTCVR improvement and 1.09% CTR improvement. These improvements are very important to Meituan hotel business. The online metric improvements can answer the experimental objective O4: DCIN can achieve better online business metrics than baselines.

5.2.4. Comparisons of computational cost.

To verify the experimental objective O5, we compare the computational cost of all models in Table 10. We find that 1) the Step/sec metrics of DIEN++ and BST++ are much lower than that of other models. This is because they need to model the relation between the click and all other clicks in the same sequence, and their time complexity is $O(n^2)$, while the other models are $O(n)$; 2) the computational cost of DCIN is slightly higher than that of DIN++ and SIM++. The slight cost gain does not significantly reduce the online efficiency. Based on the comparisons of the online metrics, we have deployed DCIN to the e-commerce platform of Meituan-Hotel with main traffic.

6. Conclusion

In this paper, we are the first to explore the interest modeling task in the reranking stage of hotel search, and identify the two key challenges, i.e., high behavior sparsity and large interest gap. To address the first challenge, we propose the deep click concept to model the multi-view semantics of a click, and then enrich the semantics of both a click and a click sequence. To address the second challenge, we develop the mutual attention interest calibration unit to calibrate the embedding of a click, and the multi-attention interest aggregating unit to estimates the weights of the calibrated embeddings from multiple perspectives. By experiments, we find 1) Interest modeling solutions of product recommendation can not work well in hotel search, because they encounter high behavior sparsity challenge and ignore the query context which is very important to hotel search; 2) Not all features of a click are suitable for representing the user's interests, such as the context features and feedback features, because user's interests are in attributes and contents of a hotel.

CRediT authorship contribution statement

Denghao Ma: Conceptualization of this study, Methodology, Software. **Hongbin Pei:** Conceptualization of this study, Methodology. **Yueguo Chen:** Conceptualization of this study, Methodology. **Peng Bao:** Conceptualization of this study, Methodology. **Liang Shen:** Conceptualization of this study, Methodology.

References

Abdool, M., Haldar, M., Ramanathan, P., Sax, T., Zhang, L., Manaswala, A., Yang, L., Turnbull, B.C., Zhang, Q., Legrand, T., 2020. Managing diversity in airbnb search, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 2952–2960. URL: https://doi.org/10.1145/3394486.3403345, doi:10.1145/3394486.3403345.

Adamczak, J., Deldjoo, Y., Moghaddam, F.B., Knees, P., Leyson, G.P., Monreal, P., 2021. Session-based hotel recommendations dataset: As part of the ACM recommender system challenge 2019. ACM Trans. Intell. Syst. Technol. 12, 1:1–1:20. URL: https://doi.org/10.1145/3412379, doi:10.1145/3412379.

Ambolkar, R., Bhagat, A., Buga, B., Gharat, S., 2022. Hotel recommendation system using advanced efficiency and accuracy with modified bert technique, in: 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp. 1747–1752. doi:10.1109/ICAIS53314.2022.9742849.

Antognini, D., Faltings, B., 2020. Hotelrec: a novel very large-scale hotel recommendation dataset, in: Proceedings of The 12th Language Resources and Evaluation Conference, European Language Resources Association. pp. 4917–4923. URL: https://aclanthology.org/2020.lrec-1.605/.

Cao, Y., Zhou, X., Huang, P., Xiao, Y., Chen, D., Chen, S., 2022. Contrastive information transfer for pre-ranking systems. CoRR abs/2207.03073. URL: https://doi.org/10.48550/arXiv.2207.03073, doi:10.48550/arXiv.2207.03073.

Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W., 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. URL: https://doi.org/10.1145/3326937.3341261.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 785–794. URL: https://doi.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y., 2015. Gated feedback recurrent neural networks, in: Proceedings of the 32nd International Conference on Machine Learning, JMLR.org. pp. 2067–2075. URL: http://proceedings.mlr.press/v37/chung15.html.

Cozza, V., Petrocchi, M., Spognardi, A., 2018. Mining implicit data association from tripadvisor hotel reviews, in: Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference, CEUR-WS.org. pp. 56–61. URL: http://ceur-ws.org/Vol-2083/paper-09.pdf.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognit. Lett. 27, 861–874. URL: https://doi.org/10.1016/j.patrec.2005.10.010, doi:10.1016/j.patrec.2005.10.010.

Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K., 2019. Deep session interest network for click-through rate prediction, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, ijcai.org. pp. 2301–2307. URL: https://doi.org/10.24963/ijcai.2019/319, doi:10.24963/ijcai.2019/319.

Findler, N.V., 1972. Short note on a heuristic search strategy in long-term memory networks. Inf. Process. Lett. 1, 191–196. URL: https://doi.org/10.1016/0020-0190(72)90037-3, doi:10.1016/0020-0190(72)90037-3.

Ghose, A., Yang, S., 2008. An empirical analysis of sponsored search performance in search engine advertising, in: Proceedings of the International Conference on Web Search and Web Data Mining, ACM. pp. 241–250. URL: https://doi.org/10.1145/1341531.1341563, doi:10.1145/1341531.1341563.

Gu, Y., Ding, Z., Wang, S., Zou, L., Liu, Y., Yin, D., 2020. Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems, in: The 29th ACM International Conference on Information and Knowledge Management, ACM. pp. 2493–2500. URL: https://doi.org/10.1145/3340531.3412697, doi:10.1145/3340531.3412697.

Haldar, M., et al., 2020. Improving deep learning for airbnb search , 2822–2830.

Hsu, C., Chung, H., Huang, H., 2003. The hybrid poisson aspect model for personalized shopping recommendation, in: Proceedings of the 3rd IEEE International Conference on Data Mining, IEEE Computer Society. pp. 545–548. URL: https://doi.org/10.1109/ICDM.2003.1250973, doi:10.1109/ICDM.2003.1250973.

Kang, W., McAuley, J.J., 2018. Self-attentive sequential recommendation, in: IEEE International Conference on Data Mining, IEEE Computer Society. pp. 197–206. URL: https://doi.org/10.1109/ICDM.2018.00035, doi:10.1109/ICDM.2018.00035.

Kaya, B., 2020. Hotel recommendation system by bipartite networks and link prediction. J. Inf. Sci. 46. URL: https://doi.org/10.1177/0165551518824577, doi:10.1177/0165551518824577.

Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L., 2019. Multi-interest network with dynamic routing for recommendation at tmall, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, ACM. pp. 2615–2623. URL: https://doi.org/10.1145/3357384.3357814, doi:10.1145/3357384.3357814.

Li, X., Zhou, X., Xiao, Y., Huang, P., Chen, D., Chen, S., Xian, Y., 2022. Autofas: Automatic feature and architecture selection for pre-ranking system. CoRR abs/2205.09394. URL: https://doi.org/10.48550/arXiv.2205.09394, doi:10.48550/arXiv.2205.09394.

Liu, B., Fu, Y., Yao, Z., Xiong, H., 2013. Learning geographical preferences for point-of-interest recommendation, in: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 1043–1051. URL: https://doi.org/10.1145/2487575.2487673, doi:10.1145/2487575.2487673.

Liu, H., Lu, J., Zhao, X., Xu, S., Peng, H., Liu, Y., Zhang, Z., Li, J., Jin, J., Bao, Y., Yan, W., 2020. Kalman filtering attention for user behavior modeling in CTR prediction, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems. URL: https://proceedings.neurips.cc/paper/2020/hash/68ce199ec2c5517597ce0a4d89620f55-Abstract.html.

Ludewig, M., Jannach, D., 2019. Learning to rank hotels for search and recommendation from session-based interaction logs and meta data, in: Proceedings of the Workshop on ACM Recommender Systems Challenge, ACM. pp. 5:1–5:5. URL: https://doi.org/10.1145/3359555.3359561, doi:10.1145/3359555.3359561.

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H., 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 1930–1939. URL: https://doi.org/10.1145/3219819.3220007, doi:10.1145/3219819.3220007.

Mavalankar, A.A., Gupta, A., Gandotra, C., Misra, R., 2019. Hotel recommendation system. CoRR abs/1908.07498. URL: http://arxiv.org/abs/1908.07498.

Pi, Q., Bian, W., Zhou, G., Zhu, X., Gai, K., 2019. Practice on long sequential user behavior modeling for click-through rate prediction, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 2671–2679. URL: https://doi.org/10.1145/3292500.3330666, doi:10.1145/3292500.3330666.

Pi, Q., Zhou, G., Zhang, Y., Wang, Z., Ren, L., Fan, Y., Zhu, X., Gai, K., 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction, in: The 29th ACM International Conference on Information and Knowledge Management, ACM. pp. 2685–2692. URL: https://doi.org/10.1145/3340531.3412744, doi:10.1145/3340531.3412744.

Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P., 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, ACM. pp. 130–137. URL: https://doi.org/10.1145/3109859.3109896, doi:10.1145/3109859.3109896.

Ramzan, B., Bajwa, I.S., Jamil, N., ul Amin, R., Ramzan, S., Mirza, F., Sarwar, N., 2019. An intelligent data analysis for recommendation systems using machine learning. Sci. Program. 2019, 5941096:1–5941096:20. URL: https://doi.org/10.1155/2019/5941096, doi:10.1155/2019/5941096.

Ren, K., Qin, J., Fang, Y., Zhang, W., Zheng, L., Bian, W., Zhou, G., Xu, J., Yu, Y., Zhu, X., Gai, K., 2019. Lifelong sequential modeling with personalized memorization for user response prediction, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM. pp. 565–574. URL: https://doi.org/10.1145/3331184.3331230, doi:10.1145/3331184.3331230.

Sharma, Y., Bhatt, J., Magon, R., 2015. A multi-criteria review-based hotel recommendation system, in: Wu, Y., Min, G., Georgalas, N., Hu, J., Atzori, L., Jin, X., Jarvis, S.A., Liu, L.C., Calvo, R.A. (Eds.), 15th IEEE International Conference on Computer and Information Technology, CIT 2015; 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015; 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2015; 13th IEEE International Conference on Pervasive Intelligence and Computing, PICom 2015, Liverpool, United Kingdom, October 26-28, 2015, IEEE. pp. 687–691. URL: https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.99, doi:10.1109/CIT/IUCC/DASC/PICOM.2015.99.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P., 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, ACM. pp. 1441–1450. URL: https://doi.org/10.1145/3357384.3357895, doi:10.1145/3357384.3357895.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wang, Y., Wang, L., Li, Y., He, D., Liu, T., 2013. A theoretical analysis of NDCG type ranking measures, in: The 26th Annual Conference on Learning Theory, JMLR.org. pp. 25–54. URL: http://proceedings.mlr.press/v30/Wang13.html.

Wu, F., Qiao, Y., Chen, J.H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., Zhou, M., 2020. Mind: A large-scale dataset for news recommendation, in: ACL.

Zhang, K., Wang, K., Wang, X., Jin, C., Zhou, A., 2015. Hotel recommendation based on user preference analysis, in: 31st IEEE International Conference on Data Engineering Workshops, IEEE Computer Society. pp. 134–138. URL: https://doi.org/10.1109/ICDEW.2015.7129564, doi:10.1109/ICDEW.2015.7129564.

Zhang, Z., Sabuncu, M.R., 2018. Generalized cross entropy loss for training deep neural networks with noisy labels, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, pp. 8792–8802. URL: https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., Gai, K., 2019. Deep interest evolution network for click-through rate prediction, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI Press. pp. 5941–5948. URL: https://doi.org/10.1609/aaai.v33i01.33015941, doi:10.1609/aaai.v33i01.33015941.

Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K., 2018. Deep interest network for click-through rate prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 1059–1068. URL: https://doi.org/10.1145/3219819.3219823, doi:10.1145/3219819.3219823.

Zhu, H., Jin, J., Tan, C., Pan, F., Zeng, Y., Li, H., Gai, K., 2017. Optimized cost per click in taobao display advertising, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 2191–2200. URL: https://doi.org/10.1145/3097983.3098134, doi:10.1145/3097983.3098134.