DCRNet: Delayed Conversion Modeling Based Personalized Flight Itinerary Ranking Network

Maolei Huang[1] *, Zhuangzhuoran[1] *, Detao Lv[1], YuanTong Li[1], Shuhan Song[2], Yao Yu[1] †

[1]Fliggy, Alibaba Group
[2]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
{hhangmaolei.hml,nantian,detao.ldt,liyuantong.lyt,sichen.yy}@alibaba-inc.com, songshuhan@ict.ac.cn

Abstract

Over recent decades, the tourism industry has demonstrated progressive expansion, driven by advancements in aviation technologies and shifting consumer interests. In this context, online flight itinerary ranking has become a pivotal business for Online Travel Platforms (OTPs), which aim to rank flight itineraries by synthesizing real-time flight data provided by airlines with users' individual travel preferences. Currently, most OTPs rely on rule-based methodologies or rudimentary user preference-driven models to address this task. However, these methods are inherently limited by their insufficient consideration of delayed booking behaviors and their neglect of dynamic contextual attributes associated with flight itineraries, thereby undermining their ability to effectively handle the intricacies of flight ranking. To address these shortcomings, this paper introduces the Delayed Conversion Modeling based Personalized Flight Itinerary Ranking Network (DCRNet), designed to improve ranking accuracy by integrating delayed booking patterns and contextual dependencies into the modeling framework. Specifically, DCRNet explores the dynamic associations between users' current contextual information and their historical travel records, and models users' delayed booking behaviors via a masked attention mechanism. Moreover, an enhanced multi-task learning framework is employed to effectively integrate traditional behavioral modeling with delay-aware modeling, thereby improving the overall prediction accuracy and enhancing the system's personalized recommendation capabilities. Extensive offline experiments conducted on real-world datasets from Amadeus and Fliggy demonstrate the superior performance of DCRNet. Furthermore, its successful deployment on Fliggy's online itinerary search system has yielded significant improvements, underscoring its practical effectiveness and scalability.

Introduction

Currently, with the gradual prosperity of the economy and the gradual improvement of people's living standards, more and more people like to relax themselves by traveling in their leisure time. Unlike before, online travel apps are becoming the most prevalent channel for booking various types of itineraries (e.g, flight, hotel and train). In our case, Fliggy [1], one of the most popular online travel apps in China, one of its core businesses is to provide users with suitable flight itineraries to meet users' need through ranking and recommendation algorithms, thereby improving user experience and improving indicators the conversion rate (CVR).

Figure 1: An Example of Itinerary Ranking at Ctrip and Expedia.

In the tourism industry, compared with traditional e-commerce ranking systems or recommendation systems (Ji et al. 2021; Sun et al. 2024) that deploy various deep learning models, the recommendation and ranking of flight itineraries still poses great challenges. At present, the most commonly used strategy of mainstream OTP platforms is to ranking itineraries based on price or duration. Figure 1 shows an example of Ctrip [2], Expedia [3] and Fliggy, where Ctrip sorts by price and Expedia sorts by flight duration by default.The main challenges of the current flight schedule sorting system are as follows:

- Dynamic Context-Aware Information and Mutual Influence: Owing to the underlying global supply and demand mechanisms, airline tickets unlike traditional e-commerce products exhibit high price volatility, with approximately 80% undergoing up to 10 price changes within a single hour. Consequently, when users issue

*These authors contributed equally.
†Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Fliggy: https://www.fliggy.com/
[2]Ctrip: https://www.ctrip.com/
[3]Expedia: https://www.expedia.com/

identical queries at different times, the contextual information associated with their preferred flights may vary substantially. Effectively modeling such dynamic context information and its mutual influence is therefore a critical challenge.

- Processing of Delayed Signals: In contrast to traditional e-commerce ranking systems, delayed feedback is prevalent in itinerary recommendation scenarios. Following a user's query, the flight search engine retrieves real-time information from global distribution system (GDS) providers and initially presents the results in a default order. Due to significant price volatility, users typically refrain from making immediate purchases even after expressing interest; Instead, they tend to monitor price trends over time, frequently revisiting the platform or comparing prices across different applications. However, existing algorithms inadequately capture this delayed conversion behavior, which leads to suboptimal online performance.

To address the above two limitations, we proposed DCRNet, a delayed conversion modeling based personalized Flight itinerary ranking network. In DCRNet, our major contributions are summarized as follows:

- To the best of our knowledge, this work constitutes the first formal definition of the delayed conversion based flight itinerary ranking problem as observed in real-world scenarios.
- We propose DCRNet, a personalized flight itinerary ranking network based on delayed conversion modeling. Within DCRNet, a series of optimization strategies are introduced, including the exploitation of key correlations between current information (e.g., query, context, target itinerary) and users' historical behavioral data, as well as the explicit modeling of delayed booking behaviors through a delay mask attention mechanism. These enhancements collectively improve the system's accuracy and its capacity for personalized recommendations.
- Empirical results from offline experiments on production datasets and online A/B testing consistently demonstrate that the proposed DCRNet significantly outperforms all baseline methods across multiple evaluation metrics. DCRNet has been successfully deployed on the Fliggy platform, where it delivers substantial improvements in itinerary reservation services for millions of users daily.

Related Works

CTR Prediction. Click-through rate (CTR) prediction has witnessed rapid development in recent years, driven by progress in both single-task and multi-task learning paradigms. Most existing CTR prediction studies primarily focus on modeling within a single scenario from several key perspectives: (1) feature interaction modeling (e.g., FM (Rendle 2010), DeepFM (Guo et al. 2017)), and (2) modeling user historical behaviors (e.g., DIN (Zhou et al. 2018), DIEN (Zhou et al. 2019)). Factorization Machines (FM) were proposed to explicitly capture pairwise feature interactions, addressing limitations of earlier linear models such as Logistic Regression (LR (LaValley 2008)) and Follow-the-Regularized-Leader (FTRL (Ta 2015)), which lack the capacity to model such interactions. Further, architectures like Wide&Deep(Cheng et al. 2016) and DeepFM jointly leverage low-order and high-order feature representations to enhance prediction accuracy. To capture fine-grained user interests, DIN introduces attention mechanisms that adaptively highlight relevant user behaviors with respect to each candidate item, while DIEN utilizes gated recurrent units (GRU (Dey and Salem 2017)) to model the dynamic evolution of user interests over time. Recognizing the limitation of representing user interests with a single vector, DMIN (Zhang et al. 2024) adopts a specially designed extractor layer to identify multiple diverse user intent vectors.

In parallel, multi-task learning (MTL) has been increasingly embraced in CTR and related tasks to exploit shared representations and improve model generalization. Notably, MMoE (Ma et al. 2018a) presents a multi-gate mixture-of-experts architecture to flexibly share knowledge across tasks. ESMM (Ma et al. 2018b) further extends MTL to capture user action modeling over the entire action space. To better disentangle shared and task-specific information, PLE (Tang et al. 2020) proposes a progressive layered extraction framework. More recently, HoME (Wang et al. 2024) and DTN (Bi et al. 2024) have introduced advanced expert-sharing and dual transfer mechanisms, which further enhance performance in large-scale multi-task learning scenarios.

Itinerary Ranking and Recommendation. In recent years, deep learning-based approaches for itinerary ranking and recommendation have achieved substantial progress, primarily driven by the development of advanced neural architectures. State-of-the-art methods include RNN-based models, such as DCM (Mottini and Acuna-Agost 2017), which is widely adopted for airline itinerary choice prediction, as well as Transformer-based frameworks exemplified by the Personalized Re-ranking Model (RPM) (Pei et al. 2019) deployed at Taobao. Notably, PFRN (Huang et al. 2020) leverages a list-wise neural ranking strategy to directly capture user preferences over candidate itineraries. Models such as PlanRanker and FitNet further improve ranking effectiveness by integrating monotonic optimization strategies and contrastive learning techniques, respectively. More recently, approaches like DCIEN (Tao et al. 2022), Hydra (Liu et al. 2019), and CPNet (Huang et al. 2025) have advanced the field by incorporating sophisticated context modeling, hierarchical structures, and collaborative preference learning, thereby setting new benchmarks for itinerary ranking and recommendation tasks.

Preliminaries

In this section, we first introduce several fundamental definitions and then formally delineate the problem that is the focus of this study.

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ denote a set of $|\mathcal{U}|$ users, and let $\mathcal{L}$ represent a set of $|\mathcal{L}|$ cities. We formally define several key concepts as follows.

Definition 1.1. Flight Itinerary. A flight itinerary **ODTF** $i(o, d, t, f) \in \mathcal{I}$, is consists of the origin city $o \in \mathcal{L}$, destination city $d \in \mathcal{L}$, departure time t and flight number f. For example, the ODTF "BJS-SIN-20251202-MU5301" indicates that the user traveled on flight MU5301 on December 2, 2025, with a departure from Beijing (BJS) and an arrival in Singapore (SIN).

Definition 1.2. Origin-Destination-Time Query. An **ODT** query, denoted as $q(o, d, t) \in \mathcal{Q}$, represents a query initiated by a user $u \in \mathcal{U}$ concerning an inter-regional flight itinerary. In this context, o and d, both elements of the set of cities $\mathcal{L}$, represent the origin and destination cities of the journey, respectively. Additionally, $t \in \mathcal{T}$ specifies the planned departure date on which the user intends to leave the origin city o. For example, the ODT query "BJS-SIN-20251202" indicates that the user queried for itineraries from Beijing (BJS) to Singapore (SIN) on the Fliggy on December 2, 2025.

Definition 1.3. User Historical Behavior Sequence. $\mathcal{H} = \{h_1, h_2, \ldots, h_{|\mathcal{H}|}\}$ represents the user's historical behavior sequence of flight itinerary, where $|\mathcal{H}|$ represents the length of the user's historical behavior. For $\forall k \in \{1, .., |\mathcal{H}|\}$, the k^{th} flight itinerary can be represented by a triple $h_k = (h_k^{id}, h_k^q, h_k^c)$, where h_k^{id} represents the flight itinerary, h_k^q represents the odt query, and h_k^c represents the context information.

With the above notations and concepts, the problem investigated in this paper is formalized as follows:

Definition 1.4. Personalized Ranking of Flight Itinerary. Given an ODT query $q(o, d, t) \in \mathcal{Q}$ initiated by a user $u \in \mathcal{U}$, and let c denotes the environmental context at that time., let $\mathcal{H} = \{(h_1^{id}, h_1^q, h_1^c), \ldots, (h_{|\mathcal{H}|}^{id}, h_{|\mathcal{H}|}^q, h_{|\mathcal{H}|}^c)\}$ denote the historical behavior sequence of flight itinerary of user u on Fliggy. Given a list of target flight itinerary (ODTFs) for the query q, denoted by $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$, the objective of flight itinerary ranking problem is to predict the probability $P(i_k|u, q, c, \mathcal{H})$ of the i^{th} target flight itinerary plan chosen by user u.

DCRNet

Overview

Figure 2 shows an overview of DCRNet. It mainly contains four components, namely Embedding Layer (**EL**), Mixed Similarity Attention Layer (**MSAL**), Delay Mask Attention Layer (**DMAL**), and Merge Layer (**ML**). EL generates embeddings for all inputs (e.g., user infos, user's historical behaviors, target ODTF and context infos), MSAL mainly obtains user's interest in the target ODTF by decoupling query and context information. DMAL aims to capture the user's interest migration by modeling the user's behavior in the delayed booking window. Then, ML uses a multi-task learning framework to dynamically fuse the two parts of the prediction.

Embedding Computation

As illustrated by Figure 2, mainly four categories of inputs: user profiles u, an ODT query q from user u, the user's historical behavior sequence $\mathcal{H}$, target itinerary t and target itinerary's context information c, are used to train DCRNet model, and each input category is represented by a group of features.

Each feature within these categories is initially transformed into a high-dimensional sparse vector and subsequently embedded into a lower-dimensional dense representation following the methodology outlined in [19]. Post-embedding, we denoted the representations as $e_u \in \mathbf{R}^d$, $e_q \in \mathbf{R}^d$ and $e_c \in \mathbf{R}^d$ for the user, query and context, respectively, with d signifying the embedding dimension. User's historical behavior sequence are encode as matrices: $\mathbf{E}_{\mathcal{H}} = \{..., e_{h_k}, ...\} \in \mathbf{R}^{|\mathcal{H}|*d}$, where $|\mathcal{H}|$ denote the length of behaviors, and for the k^{th} historical behavior in $\mathcal{H}$, represented by the triple embedding set $e_k = (e_k^q, e_k^c, e_k^{id}), e_k^q \in \mathbf{R}^d, e_k^c \in \mathbf{R}^d, e_k^{id} \in \mathbf{R}^d$ for the query, context and itinerary embeddings. Feature granularity within these embedding is maintained, with each feature's embedding divided equally across the total dimensionality for granular representation.

Mixed Similarity Attention

In our Mixed Similarity Attention Layer (MSAL), drawing inspiration from models like SIM (Pi et al. 2020), DIN and UID-NET (Lou et al. 2024), and tailored to the context of the itinerary ranking, we've enhanced the attention mechanism by devising Query-Context Extraction Module (QCEM) and Query Context Decomposition Module (QCDM) to capture the influencing factor of the current item from the user's historical behavior sequence.

QCEM is designed to extract query-context-relevant and query-context-irrelevant interests from user's historical behaviors. We first apply Multi-head Self Attention to model the interdependencies among $\mathcal{H}$. For each behavior triplet (e_k^q, e_k^c, e_k^{id}), we compute the similarity between (e_k^q, e_k^c) and the current query e_q and context e_c using attention-like mechanism, yielding the weight $w_k^{q,c}$ through Equation 1. Based on Equation 2, we then derive the query-context-relevant embedding e_k^r and query-context-irrelevant embedding e_k^{ir}. Finally, we perform target attention calculations on e_k^r and e_k^{ir} to extract information relevant to the current ODTF being scored, resulting in the refined representations e^r and e^{ir}.

$$w_k^{q,c} = \frac{(e_q e_c) \cdot (e_k^q e_k^c)}{\sum_{i=1}^{|\mathcal{H}|} (e_q e_c) \cdot (e_i^q e_i^c)} \tag{1}$$

$$\begin{aligned} e_k^r &= w_k^{q,c} \cdot e_k^{id} \cdot e_t \\ e_k^{ir} &= (1 - w_k^{q,c}) \cdot e_k^{id} \cdot e_t \end{aligned} \tag{2}$$

Then in QCDM, we employ a self-supervised learning approach to decouple the query-context-relevant and query-context-irrelevant interest embeddings. The specific methodology is as follows: we apply average pooling operations on $E^r = \{e_1^r, \ldots, e_{|\mathcal{H}|}^r\}$ and $E^{ir} = \{e_1^{ir}, \ldots, e_{|\mathcal{H}|}^{ir}\}$ to derive their respective interest proxies, denoted as p^r and p^{ir}. Subsequently, guided by the principle of contrastive learning, we treat these proxies as labels and enforce the similarity between each interest representation vector and its corresponding proxy, as formalized in Equation 3. To

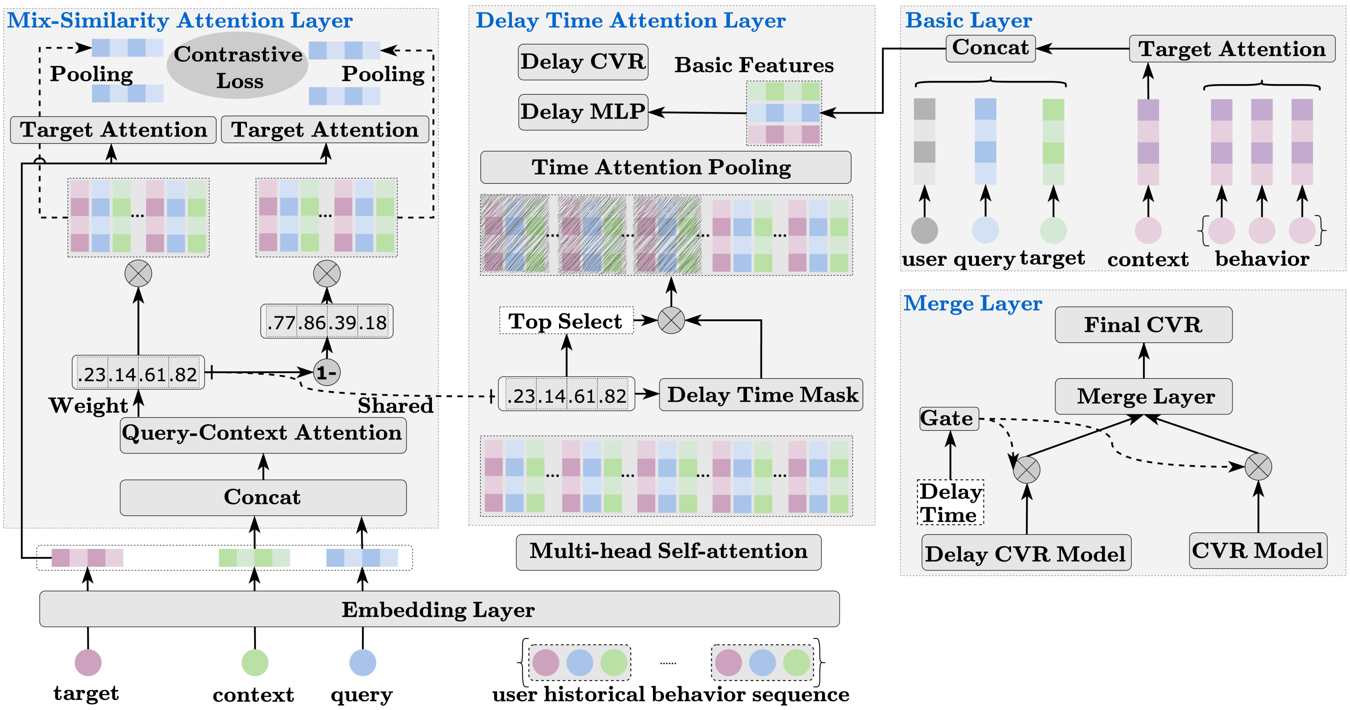

Figure 2: Framework of the DCRNet model.

impose this relationship, we utilize the Bayesian Personalized Ranking (BPR) loss function, which leverages the inner product to measure the similarity between different vectors. The detailed computation formula is presented in Equation 4.

$$\begin{aligned} \textbf{sim}(e^r, p^r) > \textbf{sim}(e^r, p^{ir}) \\ \textbf{sim}(e^{ir}, p^{ir}) > \textbf{sim}(e^{ir}, p^r) \end{aligned} \tag{3}$$

$$\mathcal{L}_{bpr} = \sigma([e^r, p^r] - [e^r, p^{ir}]) + \sigma([e^{ir}, p^{ir}] - [e^r, p^r]) \tag{4}$$

where $\sigma(\cdot)$ represents the softplus activation function, and $[.,.]$represents the inner product between two vectors.

Finally, as is shown in Equation 5 we concat e^r, e^{ir} as the output of the Mixed Similarity Attention Layer.

$$\mathcal{O}_{msa} = e^r || e^{ir} \tag{5}$$

Delay Mask Attention

As illustrated in Figure 3, for all samples $\mathcal{D}$, we categorize them into two distinct groups: $\mathcal{D}_{cbs}$, clicked before sample (CBS) and $\mathcal{D}_{ncbs}$, no clicked before sample (NCBS). This classification is based on whether the user has interacted with the same ODTF itinerary sample within a specific time window prior to the booking event, and we formally define this specific time window as the delay booking time (DBT).

In order to better model the samples in $\mathcal{D}_{cbs}$, $\mathcal{D}_{cbs}$ is further divided into $\mathcal{D}_{cbs}^{pos}$ and $\mathcal{D}_{cbs}^{neg}$, as follows: when a user subsequently have booking event, the sample resides within the data space $\mathcal{D}_{cbs}^{pos}$. In this case, we classify the sample as

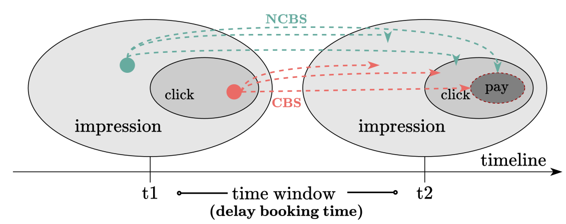

Figure 3: Clicked Before Sample (CBS) and No Clicked Before Sample(NCBS).

a delayed booking sample (DBS), and its corresponding label is designated as the delayed booking label. Conversely, if the aforementioned condition is not satisfied, the sample is referred to as a non-delayed booking sample within $\mathcal{D}_{cbs}^{neg}$.

To enhance the model's sensitivity to user behavior occurring between the last click and the subsequent purchase, we introduce a maximum delay booking time (MDT) to truncate the attention output. User sub-behavior sequences within this delay window are particularly influential in determining the likelihood of a subsequent booking. To effectively capture this phenomenon, we propose a Delay Booking Module (DBM) specifically designed to model potential shifts in user interests within this critical period. Within the DBM framework, we incorporate a delay mask attention mechanism, as formalized in Equation 6, ensuring that temporal attention pooling assigns greater weight to more recent sub-behavior sequences. These sequences are selected based on their alignment with both the query and context of historical itineraries, while also maintaining high similarity to the query and context of the target itinerary, as described in Equation 7, where the constant **c** is the top-n's weight in $W^{q,c}$.

$$\mathcal{M}' = \begin{cases} 0\ ,\ if\ \ BDT > MDT \\ 1\ ,\ if\ BDT \leq MDT \end{cases} \tag{6}$$

$$\mathcal{M}'' = \begin{cases} 0\ ,\ if\ \ w^{q,c} \leq \mathbf{c} \\ 1\ ,\ if\ \ w^{q,c} > \mathbf{c} \end{cases} \tag{7}$$

As illustrated in Equation 8, by applying the aforementioned two masking operations, we extract a sub-behavior sequence $\mathbf{E}'_{\mathcal{H}}$ from the user's historical behavior sequence, which closely resembles the user's current contextual information within the maximum delayed booking time. Subsequently, to evaluate the influence of itineraries within this sub-sequence on the target itinerary, we employ a temporal attention pooling mechanism to determine the corresponding contribution weights, as depicted in Equation 9. Here, W^t, W^i, and z denote learnable parameters.

$$\begin{aligned} \mathbf{E}'_{\mathcal{H}} &= \mathcal{M}' \cdot \mathcal{M}'' \cdot \mathbf{E}_{\mathcal{H}} \\ &= \{\ldots; e'_{h_k}; \ldots\} \end{aligned} \tag{8}$$

$$\begin{aligned} \alpha_i &= \frac{\exp(a_i)}{\sum_{j=1}^{|\mathbf{E}'_{\mathcal{H}}|} \exp a_j} \\ a_i &= z^T \mathbf{tanh}(W^t e_t + W^i e'_{h_i}) \end{aligned} \tag{9}$$

Finally, as depicted in Equations 10 and 12, a weighted sum pooling operation is applied to the matrix $\mathbf{E}'_{\mathcal{H}}$, converting it into the feature vector $\mathcal{O}_{dma}$. This transformation produces a refined representation of the user's historical behavior sequence that is highly pertinent to the target itinerary. Subsequently, the vectors $\mathcal{O}_{dma}$ and $\mathcal{O}_{msa}$ are concatenated with the embedding of $\mathcal{O}_{basic}$ (Equation 11, where $a(\cdot)$ is feed forward network with output as the activation weight), the latter being generated via a deep interest network, thereby constructing a comprehensive feature vector for subsequent modeling.

$$\mathcal{O}_{dma} = \sum_{i=1}^{|\mathbf{E}'_{\mathcal{H}}|} (\alpha_i) e'_{h_i} \tag{10}$$

$$\begin{aligned} \mathcal{O}_{basic} &= e_u || e_q || e_c || e_{u,H} \\ e_{u,H} &= f(\mathbf{E}_{\mathcal{H}}) = \sum_{i=1}^{|\mathcal{H}|} a(e_i, e_t) e_i \end{aligned} \tag{11}$$

$$y^{pre}_{delay} = \mathbf{MLP}(\mathcal{O}_{dma} || \mathcal{O}_{msa} || \mathcal{O}_{basic}) \tag{12}$$

Additionally, we introduce a customized loss function for delayed booking samples, which, unlike conventional loss functions, incorporates a time delay weight and a query-context similarity weight to better capture the correlation between the current transaction and the user's last click. Shorter booking delays and higher query-context similarities indicate a stronger link between purchase and preceding behavior, rendering such samples more informative for user intent modeling. Consequently, these samples are prioritized during the training of the delay booking module to enhance the model's ability to capture patterns in delayed transactions.

$$\begin{aligned} \mathcal{L}_{delay} &= \epsilon \cdot (y \log(y^{pre}_{delay}) + (1-y) \log(1 - y^{pre}_{delay})) \\ \epsilon &= \mathbf{MLP}(W^{q,c} || BDT) \end{aligned} \tag{13}$$

Merge Layer

In addition to modeling the delay booking samples separately, for the remaining samples, as shown in Equations 14 and Equations 15, we use the traditional CVR modeling method to align the modeling.

$$y^{pre} = \mathbf{MLP}(\mathcal{O}_{msa} || \mathcal{O}_{basic}) \tag{14}$$

$$\mathcal{L}_{basic} = y \log(y^{pre}) + (1-y) \log(1 - y^{pre}) \tag{15}$$

Finally we decompose the booking behavior into two distinct sub-problems: delay pay problem and non-delay pay problem. These sub-problems are subsequently integrated through a gating mechanism analogous to the Mixture of Experts (MoE) framework.

As illustrated in Figure 2, we introduce a gating structure designed to dynamically modulate the scores generated by the Delay Pay Sample Prediction Model and the Normal Prediction Model. The input to this gating structure is the delay booking time, which serves as a critical feature for distinguishing between the two models. Our objective is to leverage this gating mechanism to enable the model to learn the relative importance of the Delay CVR Model versus the Normal CVR Model through the back-propagation process, thereby enhancing the overall predictive performance.

Training & Serving

All parts of the modules in DCRNet are trained joint by back-propagation. In the training phase, we use the itinerary booked by the users as the labels and minimize the following loss function:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{bpr} + \alpha_2 \mathcal{L}_{delay} + \alpha_3 \mathcal{L}_{basic} \tag{16}$$

where α_1, α_2 and α_3 are learnable parameters satisfying $\alpha_1 + \alpha_2 + \alpha_3 = 1$. In Equation 16, $\mathcal{L}_{bpr}$, $\mathcal{L}_{delay}$ and $\mathcal{L}_{basic}$ are the loss values of the BPR loss in the MSA, the delay pay loss with respect in the DMA component and the basic loss in traditional cvr modeling module.

Experiment

In this section, we evaluate the performance of the proposed DCRNet extensively.

DATASET	FLIGHT		TRAIN	
	Train	Test	Train	Test
# of samples	1,200M	100M	2,000M	200M
# of + samples	120M	10M	200M	20M
# of - samples	1,080M	90M	1,800M	180M
# of users	10M	2M	10M	3M
# of itinerary	10M	5M	20M	10M
# of cities	200	200	400	400

Table 1: Statistics of the Flight Itinerary Dataset and Train Itinerary Dataset (M-million).

Experimental Setup

Dataset. There are no public datasets that contains the information of flight itinerary (ODTF), and user's log data for browsing and booking, which are all indispensable for evaluating DCRNet. Hence, two Fliggy productivity datasets, i.e., the flight itinerary dataset and the train itinerary dataset, are used in the experiment, all of which are released online. In specific, besides the information of flight and train itinerary, the datasets also contain the user's log data collected at the Fliggy platform from the date 01/06/2025 to the date 02/08/2025, while the logs of the date 02/08/2025 are used as the testing data and the remaining logs are used as the training data. Notice that only the flight (train) itinerary are clicked and booking by users are deemed as positive (+) samples, while others are treated as negative (-) samples. Statistics of the two datasets are listed in Table 1.

Baselines. We compare DCRNet with with two categories of baselines for personalized recommendation: 1) the rule-based methods that are usually adopted by most of OTPs; 2) model-based methods.

- **Cheapest**: It ranks itineraries based on their price costs.
- **Shortest**: It ranks itineraries based on their time costs.
- **LR**: It is a well-known logistic regression model which can be used to predict the ranks of different items.
- **GBDT**: It is a scalable tree-based model commonly employed in industrial recommender and ranking tasks.
- **DIN**: A deep learning model explicitly tailored for personalized recommendation tasks, particularly within online advertising and e-commerce domains.
- **Hydra**: It represents a state-of-the-art multimodal route recommendation approach that integrates Trans2vec embeddings with GBDT techniques.
- **DCM**: It utilizes a long short-term memory (LSTM) module to capture global information from list-wise inputs for airline itinerary choice prediction; however, this approach does not incorporate user preference modeling.
- **PlanRanker**: A recent personalized deep neural network for train transfer recommendation, which employs a monotonic network to model both price and time costs between the target transfer plan and alternative options.
- **PFRN**: A novel Personalized Flight Itinerary Ranking Network (PFRN) that employs list-wise feature encoding to capture global context-aware information and model mutual influences among input instances.
- **CPNet**: A Context based Personalized Deep Network for Flight Itinerary Recommendation.

Note that DIN, Hydra, DCM, FPRN, CPNet and PlanRanker are deep neural network based methods, and Cheapest, Shortest, LR and GBDT are rule-based strategies and traditional machine learning solutions.

Other settings. We evaluate the various methods using standard metrics, namely Area Under the Curve (AUC) (Wang et al. 2013) and Normalized Discounted Cumulative Gain (NDCG). AUC quantifies the probability that a randomly selected positive instance is ranked above a negative one, while NDCG, computed according to Equation 17, is a widely adopted ranking metric in practice.

$$\mathbf{NDCG@k} = \frac{\mathbf{DCG}}{\mathbf{IDCG}}, \mathbf{DCG} = \sum_{i=1}^{|\mathcal{I}|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad (17)$$

Here, rel_i indicates the relevance of the target itinerary at rank position i for an ODT query q, and $|\mathcal{I}|$ denotes the number of candidate itineraries. IDCG represents the ideal discounted cumulative gain, i.e., the DCG computed based on the optimal ranking returned by DCRNet. NDCG is the normalized version of IDCG, with values ranging from 0 to 1, where higher values indicate better ranking performance. Similarly, a higher AUC value implies superior ranking quality, where P and N denote the sets of positive and negative samples, respectively, and $\mathbf{rank}(i)$ denotes the rank position of the i^{th} positive instance.

$$\mathbf{AUC} = \frac{\sum_{i \in \mathbf{P}} \mathbf{rank}(i) - \frac{|\mathbf{P}|(|\mathbf{P}|+1)}{2}}{|\mathbf{P}||\mathbf{N}|} \quad (18)$$

Offline Evaluation

Comparison of all methods. Experimental results that compared to different methods are shown in Table 2, based on which the following conclusions can be drawn.

- The two rule-based approaches are consistently outperformed by all model-based methods across both datasets, indicating that ranking transfer plans solely according to price or time cost is a suboptimal strategy.
- All deep learning approaches (i.e., Hydra,DCM, PFRN, PlanRanker and CPNet) consistently surpass other model-based methods (i.e., LR and GBDT) across all datasets, demonstrating the effectiveness of deep learning in enhancing ranking performance.
- Compared with all baseline methods, the proposed DCRNet consistently achieves superior performance. Specifically, DCRNet attains a 1.81% improvement in AUC, and increases NDCG@10 and NDCG@15 by an average of 2.10% and 2.66%, respectively. These improvements can be attributed to DCRNet's dedicated modeling of delayed transaction samples and its dynamic mechanism for capturing correlations in historical user behaviors, as further demonstrated in subsequent ablation studies.

Categories	Methods	Flight Dataset			Train Dataset		
		AUC	NDCG@10	NDCG@15	AUC	NDCG@10	NDCG@15
Rule-based	Cheapest	0.636	0.577	0.592	0.626	0.524	0.576
	Shortest	0.683	0.586	0.607	0.662	0.566	0.598
Model-based	LR	0.699	0.592	0.612	0.671	0.573	0.612
	GBDT	0.711	0.613	0.623	0.680	0.581	0.626
	Hydra	0.732	0.627	0.647	0.694	0.588	0.637
	DCM	0.746	0.633	0.646	0.710	0.603	0.641
	PlanRanker	0.758	0.651	0.658	0.618	0.616	0.655
	PFRN	0.766	<u>0.672</u>	0.667	0.623	0.629	0.676
	CPNet	<u>0.774</u>	0.665	<u>0.677</u>	<u>0.633</u>	<u>0.638</u>	<u>0.689</u>
	DCRNet	**0.788**	**0.686**	**0.695**	**0.657**	**0.669**	**0.701**

Table 2: Comparison of different methods on two datasets (Best values are in bold; next-best values are underlined).

-		AUC	NDCG@10	NDCG@15
w/o MSA	F	0.781	0.672	0.644
	T	0.646	0.652	0.693
w/o DMA	F	0.785	0.678	0.688
	T	0.648	0.659	0.698

Table 3: Comparison of DCRNet and its variants (F: Flight, T: Train. Best values are in bold; next best values are underlined).

Ablation Study. The ablation studies are carried out to further understand the importance of every optimization strategies proposed in DCRNet. In specific, DCRNet is compared with its three variants listed below:

- w/o MSA: The Mixed Similarity Attention (MSA) component is removed from DCRNet.
- w/o DMA: The Delay Mask Attention (DMA) component is removed from DCRNet.

The ablation study results are presented in Table 3. As demonstrated in the table, the MSA component and DMA component each play a significant role in enhancing the quality of the predicted scores for nearby flight recommendations generated by DCRNet.

Online A/B Test

Following the success of offline experiments, we deploy DCRNet in the daily production environment at Alibaba Fliggy. The online experiment involves rule-based methods (i.e., Cheapest and Shortest), as well as the best model-based baselines (i.e., PlanRanker, CPNet and DCRNet), to handle real flight traffic generated by 2 million users on Alibaba Fliggy over 10-day period in July 2025. To ensure fairness, the scheduling engine of Fliggy is adjusted to evenly distribute approximately $\frac{1}{5}$ of the daily traffic to each method.

Two key industrial metrics, Click-Through Rate (**CTR**) and Conversion Rate (**CVR**), are used to evaluate the performance of the recommendation systems. These metrics measure the effectiveness of different methods in serving online flight or train itineraries. CTR is calculated using Equation 19, while CVR is computed using Equation 20.

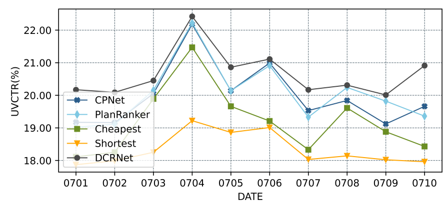

Figure 4: Online CTRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of flight itinerary ranking.

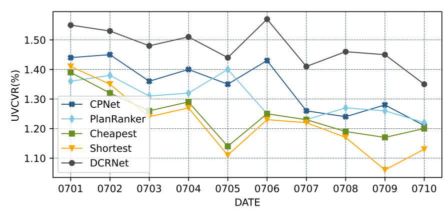

Figure 5: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of flight itinerary ranking.

$$\mathbf{CTR} = \frac{\textbf{\# number of clicked users}}{\textbf{\# number of impressed users}} \quad (19)$$

$$\mathbf{CVR} = \frac{\textbf{\# number of booking users}}{\textbf{\# number of impressed users}} \quad (20)$$

As shown in Figure 4 and Figure 5, the online experimental results indicate that the proposed DCRNet has achieved significant improvements in CTR and CVR in the scenario of flight itinerary ranking, while Figures 6 and 7 also show that DCRNet has the same significant effect in the scenario of train itinerary ranking.

Specifically, the proposed approach yields an average CTR improvement of 14.6% over the cheapest solution, which considers only the price cost of the itinerary, and 23.1% over the shortest solution, which considers only the time cost. For CVR, average improvements of 11.1% and 12.6% are observed over the Cheapest and Shortest solu-

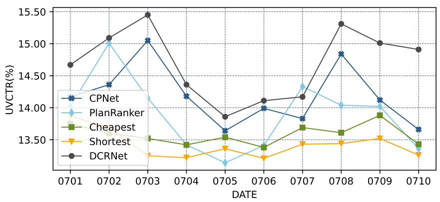

Figure 6: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of train itinerary ranking.

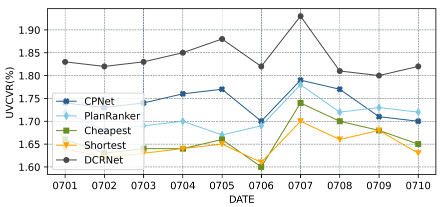

Figure 7: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of train itinerary ranking.

tions, respectively. Furthermore, DCRNet achieves an average CTR gain of 3.71% compared to the baseline PFRN and 8.03% compared to CPNet. In terms of CVR, CPNet demonstrates even greater improvements, with average gains of 11.56% over PFRN and 6.24% over CPNet. These results underscore the efficacy of DCRNet in enhancing both user engagement and conversion rates within industrial application scenarios.

Deployment and Efficiency

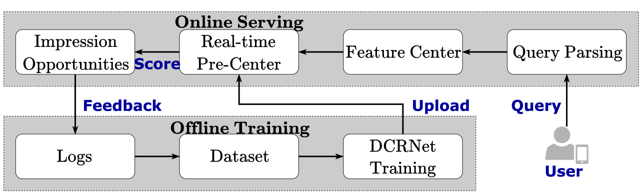

Figure 8: Illustration of deployment of DCRNet at Fliggy.

Deployment of DCRNet at Fliggy. DCRNet has been deployed at Fliggy, one of the most popular online travel platforms (OTPs) in China. It has successfully optimized the flight (train) itineraries ranking at Fliggy and serves millions of users each day. Deployment of DCRNet at Fliggy is illustrated in Figure 8. As shown by the figure, in the offline training stage, logs of user feedback are gathered which are then stored at the MaxCompute platform of Alibaba Group and the training dataset is built by processing the stored logs. After that, the DCRNet model is trained in a distributed way using Tensorflow with 20 parameter servers and 20 workers deployed at Alibaba PAI (Platform of Artificial Intelligence). Notice that each parameter server owns 6 CPU cores with 32 GB RAM, being responsible for fetching part of training samples, computing gradients of parameters, and sending the gradients to parameter servers. Then, the trained DCRNet model is uploaded to the real-time prediction (RTP) center. When a user initializes an ODT query to search for flight (train) itineraries at Fliggy, the ODT query is firstly parsed, after which the candidate itineraries, the user's profile together with historical behaviors are fetched from the feature center. According to the parsed ODT query and all fetched data, the RTP center calculates the ranking score for each candidate itineraries based on the trained DCRNet model. Finally, k itineraries with the highest ranking scores are ranked and exposed to the user in the Fliggy Mobile APP.

Method	Training Time	Inference Time
Cheapest	-	2ms
Shortest	-	2ms
GBDT	34m19s	14ms
Hydra	58m28s	27ms
PFRN	75m30s	29ms
PlanRanker	77m46s	25ms
CPNet	77m33s	26ms
DCRNet	73m28s	26ms

Table 4: Comparison of efficiency of different methods (m-minute; s-second; ms-millisecond).

Efficiency evaluation Table 4. demonstrates that there are no statistically significant differences in training time among various model-based methods when utilizing the large-scale cluster deployed at Fliggy. However, deep neural network-based approaches exhibit slightly longer training durations compared to GBDT, primarily due to their higher architectural complexity. In contrast, the Cheapest (or Shortest) method, being rule-based, does not require training and thus incurs no training overhead. Furthermore, as observed from the figure, the online inference time for all methods remains below 0.03 seconds, ensuring compliance with the stringent latency requirements typical of industrial applications.

Conclusion

Summary. This paper introduces DCRNet, a delayed conversion modeling based personalized flight itinerary ranking network. DCRNet not only models query-context-relevant and query-context-irrelevant, but also use delay mask attention to capture delay booking behaviors' information. Experimental results on large-scale real-world datasets demonstrate the strong performance of DCRNet in ranking flight and train itineraries.

Generalization. Although DCRNet is developed in the context of flight and train itinerary ranking problem, its framework can be readily adapted to other personalized ranking scenarios, such as personalized hotel ranking, and general item ranking. This is attributed to the shared need for modeling critical information and incorporating reference plans to enhance ranking effectiveness across these domains.

References

Bi, Y.; Lian, Y.; Cui, J.; Liu, J.; Wang, P.; Li, G.; Chen, X.; Zhao, J.; Wen, H.; Zhang, J.; et al. 2024. DTN: Deep Multiple Task-specific Feature Interactions Network for Multi-Task Recommendation. *arXiv preprint arXiv:2408.11611*.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, 7–10.

Dey, R.; and Salem, F. M. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, 1597–1600. IEEE.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247*.

Huang, J.; Li, Y.; Sun, S.; Zhang, B.; and Huang, J. 2020. Personalized flight itinerary ranking at fliggy. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2541–2548.

Huang, M.; Lv, D.; Yu, Y.; Song, S.; Li, D.; and Zhuang, Z. 2025. A Context based Personalized Deep Network for Nearby Flight Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 4533–4542.

Ji, H.; Zhu, J.; Wang, X.; Shi, C.; Wang, B.; Tan, X.; Li, Y.; and He, S. 2021. Who you would like to share with? a study of share recommendation in social e-commerce. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 232–239.

LaValley, M. P. 2008. Logistic regression. *Circulation*, 117(18): 2395–2399.

Liu, H.; Tong, Y.; Zhang, P.; Lu, X.; Duan, J.; and Xiong, H. 2019. Hydra: A personalized and context-aware multimodal transportation recommendation system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2314–2324.

Lou, J.; Li, Z.; Wen, H.; Lv, J.; Zhang, J.; Lv, F.; Chen, Z.; and Wu, J. 2024. UID-Net: Enhancing Click-Through Rate Prediction in Trigger-Induced Recommendation Through User Interest Decomposition. In *International Conference on Advanced Data Mining and Applications*, 49–64. Springer.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018a. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.

Ma, X.; Zhao, L.; Huang, G.; Wang, Z.; Hu, Z.; Zhu, X.; and Gai, K. 2018b. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1137–1140.

Mottini, A.; and Acuna-Agost, R. 2017. Deep choice model using pointer networks for airline itinerary prediction. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1575–1583.

Pei, C.; Zhang, Y.; Zhang, Y.; Sun, F.; Lin, X.; Sun, H.; Wu, J.; Jiang, P.; Ge, J.; Ou, W.; et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*, 3–11.

Pi, Q.; Zhou, G.; Zhang, Y.; Wang, Z.; Ren, L.; Fan, Y.; Zhu, X.; and Gai, K. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2685–2692.

Rendle, S. 2010. Factorization machines. In *2010 IEEE International conference on data mining*, 995–1000. IEEE.

Sun, J.; Ding, Z.; Chen, X.; Chen, Q.; Wang, Y.; Zhan, K.; and Wang, B. 2024. Cread: A classification-restoration framework with error adaptive discretization for watch time prediction in video recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9027–9034.

Ta, A.-P. 2015. Factorization machines with follow-the-regularized-leader for CTR prediction in display advertising. In *2015 IEEE international conference on big data (Big data)*, 2889–2891. IEEE.

Tang, H.; Liu, J.; Zhao, M.; and Gong, X. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM conference on recommender systems*, 269–278.

Tao, W.; Fu, Z.-H.; Li, L.; Chen, Z.; Wen, H.; Liu, Y.; Shen, Q.; and Chen, P. 2022. A dual channel intent evolution network for predicting period-aware travel intentions at fliggy. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 3524–3533.

Wang, X.; Cao, J.; Fu, Z.; Gai, K.; and Zhou, G. 2024. HoME: Hierarchy of Multi-Gate Experts for Multi-Task Learning at Kuaishou. *arXiv preprint arXiv:2408.05430*.

Wang, Y.; Wang, L.; Li, Y.; He, D.; and Liu, T.-Y. 2013. A theoretical analysis of NDCG type ranking measures. In *Conference on learning theory*, 25–54. PMLR.

Zhang, Z.; Li, Z.; Jin, J.; Gao, X.; Yang, X.; Zhang, B.; and Xiao, L. 2024. DeepMIN: Deep Multi-modal Interest Network with Cognitive Learning Modules. In *International Conference on Database Systems for Advanced Applications*, 212–227. Springer.

Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; and Gai, K. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5941–5948.

Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; and 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*, 1059–1068.