Association for the Advancement of Artificial Intelligence

Fliggy

# DCRNet : Delayed Conversion Modeling based Personalized Flight Itinerary Ranking Network

Maolei Huang, Zhuoran Zhuang, Detao Lv, Yuantong Li, Shuhan Song, Yao Yu *

Fliggy Alibaba Group, ICT, UCAS

## Background


Ctrip　　Expedia

The tourism industry's growth has elevated online flight itinerary ranking to a core function of Online Travel Platforms (OTPs), which must align real-time flight data with user preferences. Existing approaches largely rule-based or simplistic preference models, fail to account for delayed booking behaviors and dynamic contextual factors, limiting ranking effectiveness. To address this, we propose **DCRNet** (**D**elayed **C**onversion **M**odeling based Personalized Flight Itinerary **R**anking **Net**work), which captures temporal context through user–itinerary interactions and models delayed conversions via a masked attention mechanism. An enhanced multi-task learning framework jointly optimizes conventional behavioral signals and delay aware objectives. Evaluated on real world datasets from Amadeus and Fliggy, DCRNet achieves superior offline performance and has been successfully deployed on Fliggy's production system, delivering significant gains in ranking quality and user engagement.

## Challenge

### PRESENT STATUS

a) Enterprises: Ctrip, Fliggy, UMETRIP

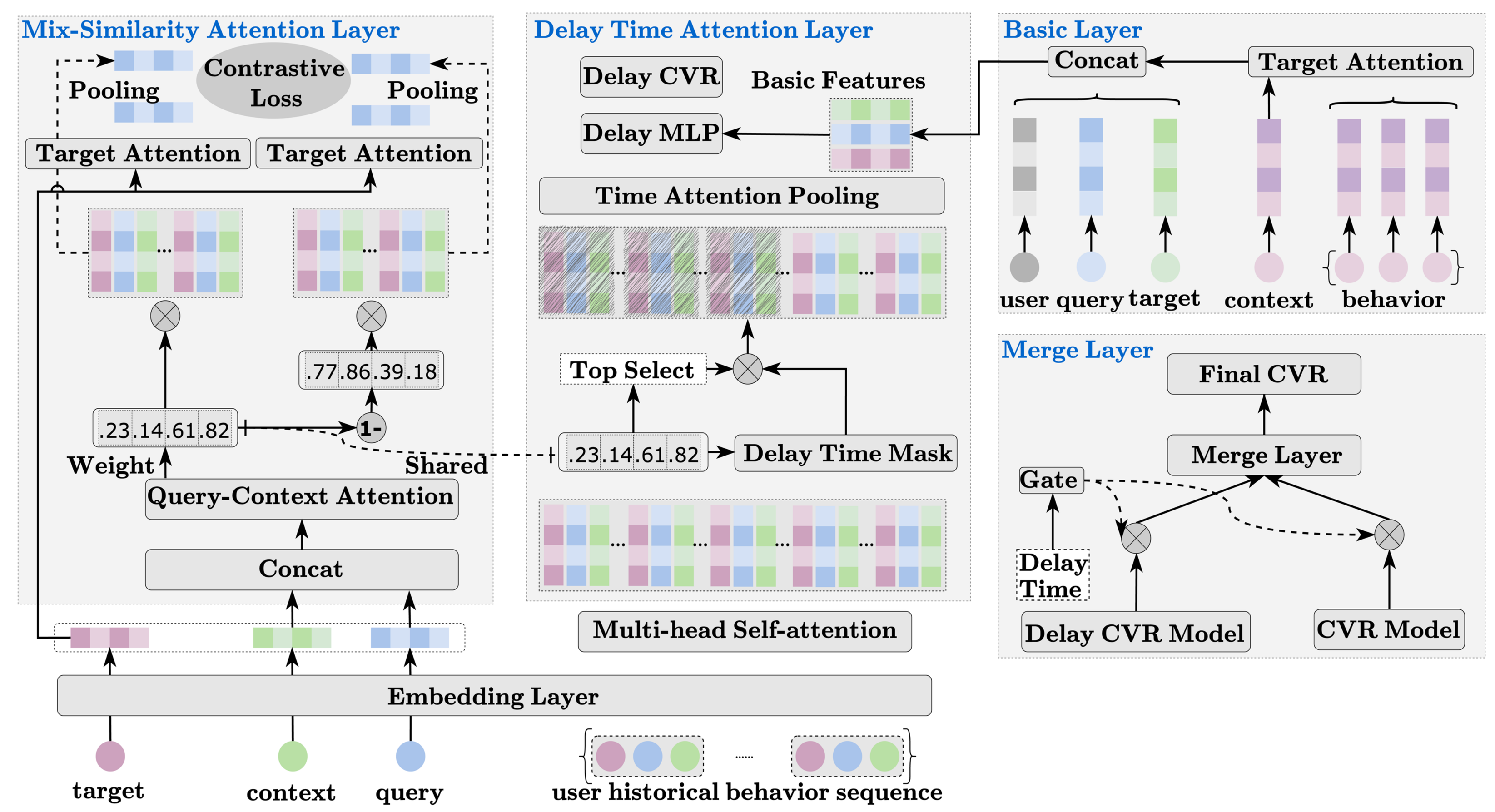
b) Methods: single-task models (e.g, DNN, DIN PFRN and CPNet) and multi-task models models (e.g, ESMM, PLE MMoE).

### LIMITATIONS

a) Airline tickets exhibit extreme price volatility 80% change price up to 10 times per hour, making contextual signals highly dynamic: identical queries at different times yield different results. Modeling this temporal context and its mutual influence is crucial.

b) Additionally, user feedback is often delayed, as travelers typically monitor prices and revisit before booking, rather than converting immediately. Existing ranking methods poorly capture this delayed conversion behavior, degrading online performance. Effectively integrating dynamic context and delayed signals remains a key challenge in travel recommendation.

## Architecture Design




### Mixed Similarity Attention

**QCEM** (Query-Context Extraction Module) is designed to extract query-context-relevant and query-context-irrelevant interests from user's historical behaviors.

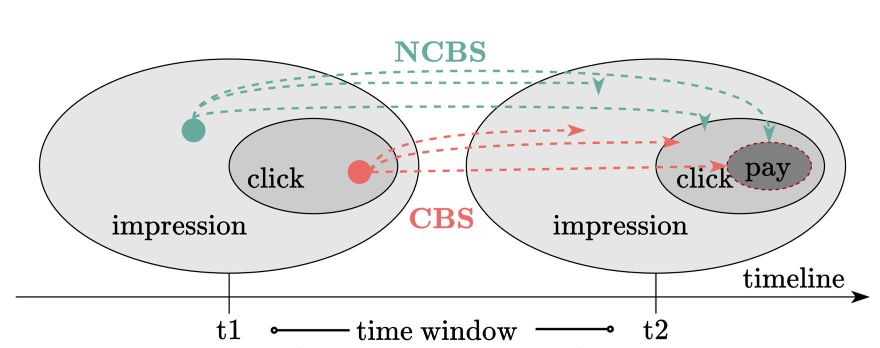
$$w_k^{q,c} = \frac{(e_q e_c) \cdot (e_k^q e_k^c)}{\sum_{i=1}^{|\mathcal{H}|}(e_q e_c) \cdot (e_i^q e_i^c)} \qquad \begin{aligned} e_k^r &= w_k^{q,c} \cdot e_k^{id} \cdot e_t \\ e_k^{ir} &= (1 - w_k^{q,c}) \cdot e_k^{id} \cdot e_t \end{aligned}$$

**QCDM** (Query Context Decomposition Module) employ a self-supervised learning approach to decouple the query-context-relevant and query-context-irrelevant interest embeddings.

$$\begin{aligned} \mathbf{sim}(e^r, p^r) &> \mathbf{sim}(e^r, p^{ir}) \\ \mathbf{sim}(e^{ir}, p^{ir}) &> \mathbf{sim}(e^{ir}, p^r) \end{aligned} \qquad \mathcal{L}_{bpr} = \sigma([e^r, p^r] - [e^r, p^{ir}]) + \sigma([e^{ir}, p^{ir}] - [e^r, p^r])$$

### Delay Mask Attention

We categorize samples into clicked before booking (**CBS**) and non-clicked (**NCBS**) groups based on user interactions within a defined delay booking time (**DBT**). CBS is further split into positive (delayed booking) and negative samples.




To capture interest shifts between last click and purchase, we propose a Delay Booking Module (**DBM**) with a maximum delay time (**MDT**) and a delay-masked attention mechanism that prioritizes recent, contextually relevant sub-behaviors. Temporal attention pooling extracts influential historical itineraries, and the resulting refined representation is fused with base features for final prediction.

$$\begin{aligned} \mathbf{E}'_{\mathcal{H}} &= \mathcal{M}' \cdot \mathcal{M}'' \cdot \mathbf{E}_{\mathcal{H}} \\ &= \{\ldots; e'_{h_k}; \ldots\} \\ \alpha_i &= \frac{\exp(a_i)}{\sum_{j=1}^{|\mathbf{E}'_{\mathcal{H}}|} \exp a_j} \\ a_i &= z^T \mathbf{tanh}(W^t e_t + W^i e'_{h_i}) \end{aligned} \qquad \begin{aligned} \mathcal{O}_{dma} &= \sum_{i=1}^{|\mathbf{E}'_{\mathcal{H}}|} (\alpha_i) e'_{h_i} \\ \mathcal{O}_{basic} &= e_u || e_q || e_c || e_{u,H} \\ e_{u,H} &= f(\mathbf{E}_{\mathcal{H}}) = \sum_{i=1}^{|\mathcal{H}|} a(e_i, e_t) e_i \end{aligned}$$

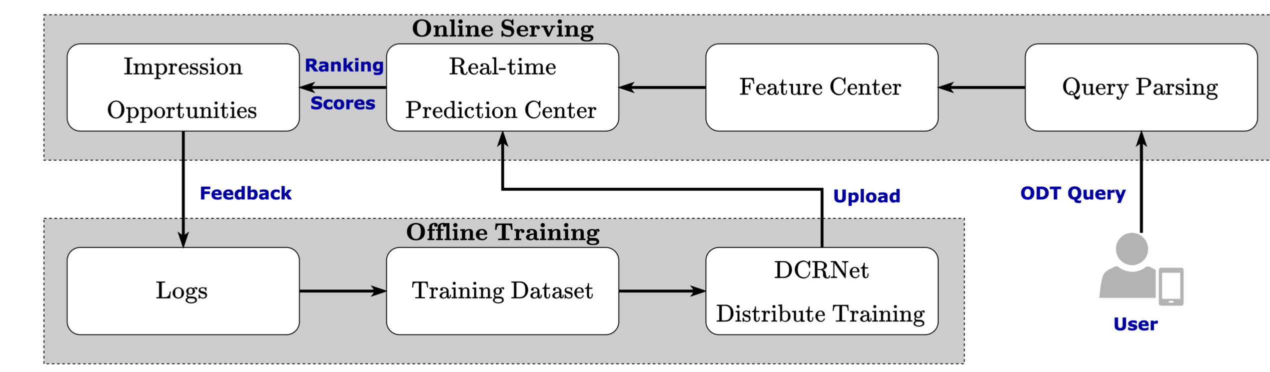
$$\begin{aligned} y_{delay}^{pre} &= \mathbf{MLP}(\mathcal{O}_{dma} || \mathcal{O}_{msa} || \mathcal{O}_{basic}) \\ \mathcal{L}_{delay} &= \epsilon \cdot (y \log(y_{delay}^{pre}) + (1-y)\log(1 - y_{delay}^{pre})) \\ \epsilon &= \mathbf{MLP}(W^{q,c} || BDT) \end{aligned}$$

### Merge Layer

We decompose booking prediction into delayed and non-delayed sub-problems, modeling them separately with dedicated CVR models. A gating mechanism, conditioned on delay booking time and trained end-to-end, dynamically combines their outputs to enhance overall performance.

$$y^{pre} = \mathbf{MLP}(\mathcal{O}_{msa} || \mathcal{O}_{basic}) \qquad \mathcal{L}_{basic} = y\log(y^{pre}) + (1-y)\log(1-y^{pre})$$

## System Deployment Design




### Deployment of DCRNet at Fliggy

DCRNet has been deployed on Fliggy, a leading online travel platform, to enable personalized ranking of flight and train itineraries for millions of daily users. User logs are collected offline, stored in Alibaba's MaxCompute. The model is trained in a distributed manner on Alibaba PAI and deployed to the Real-Time Prediction (RTP) center. During inference, an ODT query from the Fliggy app triggers the retrieval of candidate itineraries, user profiles, and historical behaviors. DCRNet computes ranking scores in real time, and the top-k results are served to the user enabling scalable, semantics aware travel recommendations.

## Offline Evaluation

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

| 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 3: Comparison of DCRNet and its variants (F: Flight, T: Train. 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 4: Comparison of efficiency of different methods (m-minute; s-second; ms-millisecond).

| 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 |

## Online A/B Test

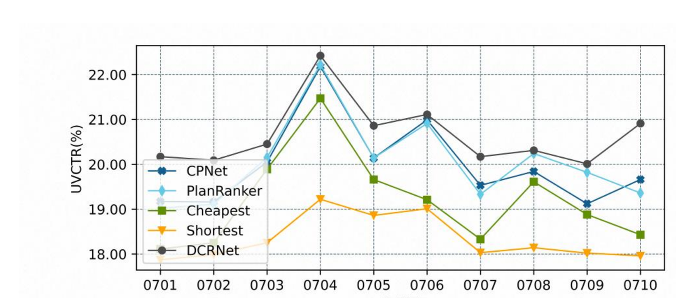


Figure 4: Online CTRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of flight itinerary ranking.

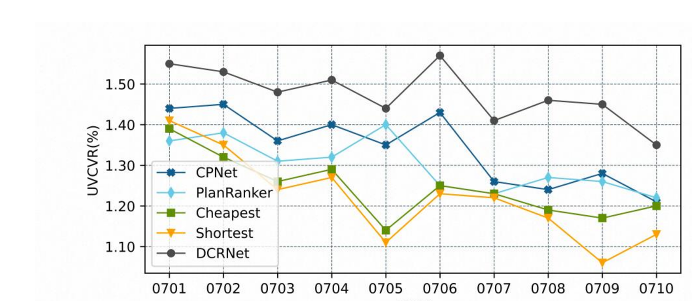


Figure 5: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of flight itinerary ranking.

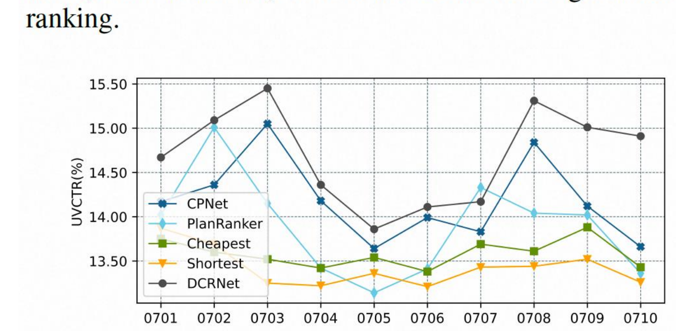


Figure 6: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of train itinerary ranking.

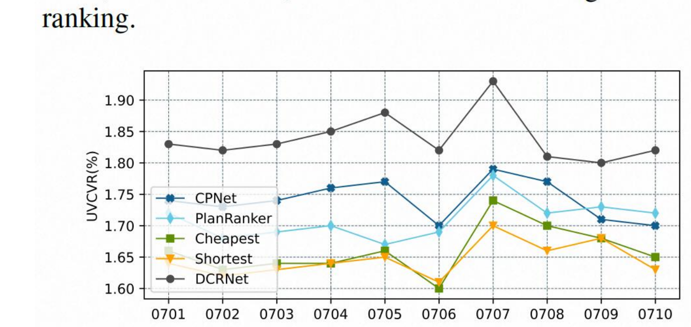


Figure 7: Online CVRs of different methods at Fliggy from Jul. 1, 2025 to Jul. 10, 2025 in the scenario of train itinerary ranking.