



 Latest updates: <https://dl.acm.org/doi/10.1145/3701716.3715250>

RESEARCH-ARTICLE

MRGRP: Empowering Courier Route Prediction in Food Delivery Service with Multi-Relational Graph

CHANG LIU, Beijing National Research Center for Information Science and Technology, Beijing, China

HUAN YAN, Beijing National Research Center for Information Science and Technology, Beijing, China

HONGJIE SUI, Beijing National Research Center for Information Science and Technology, Beijing, China

HAOMIN WEN, Carnegie Mellon University, Pittsburgh, PA, United States

YUAN YUAN, Beijing National Research Center for Information Science and Technology, Beijing, China

YUYANG HAN, Meituan, Beijing, China

[View all](#)

Open Access Support provided by:

Carnegie Mellon University

Meituan

Beijing National Research Center for Information Science and Technology



PDF Download
3701716.3715250.pdf
14 January 2026
Total Citations: 0
Total Downloads: 1266

Published: 08 May 2025

[Citation in BibTeX format](#)

WWW '25: The ACM Web Conference

2025

April 28 - May 2, 2025

Sydney NSW, Australia

Conference Sponsors:

SIGWEB

MRGRP: Empowering Courier Route Prediction in Food Delivery Service with Multi-Relational Graph

Chang Liu

Huan Yan*

Hongjie Sui

Department of Electronic

Engineering, BNRist,

Tsinghua University, Beijing, China

lc23@mails.tsinghua.edu.cn

yanhuan@tsinghua.edu.cn

Haomin Wen

Carnegie Mellon University,

Pittsburgh, PA, USA

Yuan Yuan

Department of Electronic

Engineering, BNRist,

Tsinghua University, Beijing, China

Yuyang Han

Hongsen Liao

Xuetao Ding

Jinghua Hao

Meituan, Beijing, China

Yong Li*

Department of Electronic

Engineering, BNRist,

Tsinghua University, Beijing, China

liyong07@tsinghua.edu.cn

Abstract

Instant food delivery has become one of the most popular web services worldwide due to its convenience in our daily lives. A fundamental problem in this service scenario is the accurate prediction of courier routes, which is essential for optimizing task dispatch and thereby improving delivery efficiency. It not only increases the satisfaction of both couriers and users but also drives higher profitability for the platform. The deployed heuristic prediction approach of the platform accounts for only limited human-selected task features and neglects couriers' preferences, resulting in sub-optimal performances. Moreover, existing learning-based methods fail to adequately explore the diverse factors influencing couriers' decision-making behaviors and intricate relationships among factors. To meet the urgent need for a powerful route prediction approach that benefits millions of couriers, users, and the platform itself, we propose a Multi-Relational Graph-based Route Prediction (MRGRP) method, which enables fine-grained modeling of the correlations among tasks influencing couriers' decision-making and achieves accurate prediction. We encode spatial and temporal proximity, along with the pickup-delivery relationships of tasks, into a multi-relational graph, then design a GraphFormer architecture to capture these complex correlations. Furthermore, we introduce a route decoder that leverages information about couriers as well as dynamic distance and time contexts for dynamic prediction. It also utilizes existing route solutions as a reference to find better outcomes. Experimental results demonstrate that our proposed model attains state-of-the-art performance in route

prediction on offline data from cities of diverse scales. We then deploy our model on the Meituan Turing online platform, where it significantly surpasses the existing deployed heuristic algorithm and achieves a high route prediction accuracy of 0.819, which is required for the quality and satisfaction of couriers and users in the instant food delivery service. We have open-sourced our code at <https://github.com/tsinghua-fib-lab/MGRRoute>.

CCS Concepts

- Computing methodologies → Planning for deterministic actions.

Keywords

Food delivery service; Route prediction; Multi-relational graph

ACM Reference Format:

Chang Liu, Huan Yan*, Hongjie Sui, Haomin Wen, Yuan Yuan, Yuyang Han, Hongsen Liao, Xuetao Ding, Jinghua Hao, and Yong Li*. 2025. MRGRP: Empowering Courier Route Prediction in Food Delivery Service with Multi-Relational Graph. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3701716.3715250>

1 Introduction

Instant food delivery services have become an indispensable aspect of contemporary daily life, primarily due to the convenience they offer in consumption [2, 11, 17]. After customers submit orders, the service platform communicates the order details to the restaurant and assigns the delivery task to a courier. Once the courier accepts the assignment, they collect the prepared food at the restaurant and subsequently deliver it to the customer's location. Existing analysis [6, 18, 36] indicate significant deviations between the actual routes of couriers and the recommended routes by the platform. Therefore, one of the fundamental problems in this service scenario is the accurate prediction of courier routes, which is crucial for downstream tasks, including arrival time estimation and task dispatching. Effective route prediction facilitates delivery efficiency

* Huan Yan and Yong Li are corresponding authors.



This work is licensed under a Creative Commons Attribution International 4.0 License.

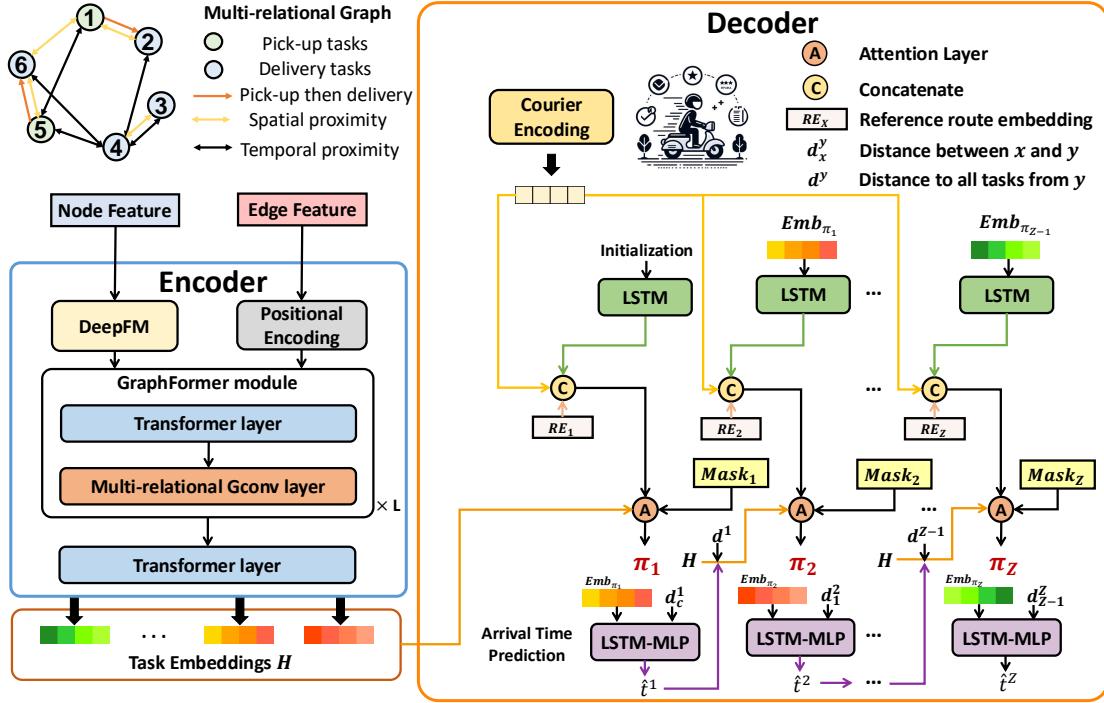


Figure 1: The overall framework of MRGRP.

and task allocation, thereby improving the working experiences of couriers and increasing user retention. Ultimately, these advancements contribute to greater profitability for the platform.

Extensive research and numerous initiatives have been undertaken in the field to address this issue. Rule-based methods, such as TimeRank and DisGreedy, predict routes based solely on distance and time, thereby neglecting the intricate features associated with tasks. The heuristic approach TSFH deployed by Meituan, China's largest food delivery service provider, considers only a limited set of human-selected features and overlooks couriers' preferences. Consequently, these methods result in sub-optimal performance and fail to meet the stringent requirements for downstream tasks. With advancements in learning techniques, significant efforts have been made to leverage these methods to capture diverse task features, thereby enhancing prediction performance. For instance, FDNET [6] utilizes a deep architecture to process task features and integrates a combination of LSTM and attention network to predict future routes. Graph2Route [34] introduces a dynamic graph neural network to capture task correlations. Although these methods have demonstrated improved prediction performance, there are still three key challenges that require further consideration:

- **Inadequate exploration of the multifaceted correlations and dependencies among tasks.** During route selection, it is crucial to simultaneously consider various aspects of task dependencies and correlations, such as spatial-temporal proximity and sequential order. For example, after completing a pick-up task, a courier may face three subsequent delivery tasks: one that is spatially proximate but not urgent, another that is distant but requires immediate delivery, and a third with moderate distance and urgency, corresponding to the same order as the

just-picked food. Accurately capturing and managing these diverse and complex factors presents a significant challenge for existing frameworks. As the number of tasks increases, their interdependencies become increasingly complex, necessitating the development of effective modules to capture these correlations and thereby facilitate route predictions.

- **Large solution search space in route prediction.** Route prediction models are usually required to identify the most executable route from a vast feasible solution space that grows exponentially with the number of tasks. Such inherent complexity renders the route prediction problem exceptionally challenging. Consequently, there is a pressing need for the development of advanced frameworks capable of effectively managing the vast solution space of route predictions.

- **Inherent dynamism of problem conditions.** The pick-up and delivery process is characterized by an inherent dynamism, where the problem context continuously evolves in response to the courier's changing location and the passage of time. Such dynamism introduces complexities that cannot be overlooked during route prediction. For instance, the relative distances between tasks shift as the courier moves to a new location. Additionally, certain tasks that may have initially appeared non-urgent can escalate in priority as time progresses, demanding immediate attention. Ignoring this spatial and temporal dynamism can lead to sub-optimal routes and potential delays in service.

To deal with the above challenges, we develop a powerful Multi-Relational Graph-based Route Prediction (MRGRP) method with an encoder-decoder architecture. To address the first challenge, we extract and establish multiple relationships among dispatched tasks

by considering various factors that influence couriers' decision-making and represent these relationships in the form of a multi-relational graph. Furthermore, we design a multi-relational graph encoder based on the GraphFormer architecture, which proficiently assimilates these intricate correlations among different tasks. To tackle the second challenge, we design a route decoder that leverages existing competitive route solutions as a reference to guide the prediction process. The decoder utilizes distances and the estimated remaining time of tasks to enhance dynamic route prediction, effectively narrowing the solution space and improving the prediction accuracy. Additionally, we introduce an auxiliary arrival time prediction procedure that estimates arrival times at each decoding step, thereby enabling the calculation of the remaining time for each task. By further integrating the distance to each task at every decoding step, we construct dynamic features that effectively address the third challenge. In summary, our main contributions are as follows:

- We propose MRGRP, a novel route prediction approach consisting of an MR-graph encoder designed to capture intricate correlations between tasks within the constructed multi-relational graph and a reference-based route decoder that efficiently narrows the solution space and enables dynamic route decoding.
- We conduct extensive offline experiments on three real-world datasets to demonstrate the superiority of MRGRP, which improves Location Square Deviation (LSD) by 17.73-20.31%, and Same Rate (SR) by 3.90-5.96% over the state-of-the-art baselines.
- We deploy our framework on Meituan, China's largest food delivery platform, and demonstrate that MRGRP outperforms the currently deployed heuristic algorithm, achieving a 31.85% reduction in LSD and a 7.46% increase in SR. These improvements are crucial for enhancing the quality of service, satisfaction of both couriers and users and, ultimately, the profitability of the instant food delivery platform of Meituan.

2 Preliminaries

2.1 Multi-relational Graph

In food delivery services, there are two types of tasks: pick-up and delivery tasks. For a specific courier c , his/her unfinished tasks and their relations can be represented with a multi-relational graph $\mathcal{G}^c = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E}, \mathcal{R})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ denotes the nodes of the graph. Each node corresponds to the courier's pick-up or delivery task. $\mathcal{R} = \{r_1, r_2, \dots, r_w\}$ represents w kinds of relations r between tasks. $\mathcal{E} = \{(u, v, r) | u, v \in \mathcal{V}, r \in \mathcal{R}\}$ denotes the edge set of the graph, representing the relation type r exist between node u and v . $\mathbf{X} \in \mathbb{R}^{n \times d_v}$ denotes node features, where d_v is its dimension. Node features are constructed according to attributes of tasks, such as their geographical location and the predicted finish time. $\mathbf{E} = \{\mathbf{E}^1, \mathbf{E}^2, \dots, \mathbf{E}^w\}$ represents the edge features for w kinds of relations, where $\mathbf{E}^r \in \mathbb{R}^{n \times n \times d_e^r}$ and d_e^r denotes the edge feature dimension for the relation r . Edge features of different relations are collected with the corresponding information between tasks. For example, the edge feature of spatial correlation can include the geographical distance between the two tasks.

2.2 Problem Formulation

Given a courier c and the input graph $\mathcal{G}^c = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E}, \mathcal{R})$ including his/her pick-up and delivery tasks and their features as well as correlations, our objective is to derive a function $\mathcal{F}_{\mathcal{P}}$ to predict the

route $\hat{\pi}$ of tasks \mathcal{V} , which is a permutation of its elements limited by route constraints \mathcal{P} :

$$\mathcal{F}_{\mathcal{P}}(\mathcal{G}^c) = \pi_1, \pi_2, \dots, \pi_{|\mathcal{V}|}, \quad (1)$$

where $\pi_i \in \{1, 2, \dots, |\mathcal{V}|\}$ denotes the predicted index of the task finished at step i , and $\pi_i \neq \pi_j$ if $i \neq j$. $|\cdot|$ represents the set cardinality. In our problem, we consider a universal pick-up then delivery constraint within \mathcal{P} , *i.e.*, the delivery of an order can only be finished after the courier finishes the pick-up task of the same order [6].

3 Methodology

In this section, we propose a multi-relational graph-based model named MRGRP (short for Multi-Relational Graph-based Route Prediction) to solve the route prediction problem for food delivery services considering the common pick-up then delivery constraint.

Suppose that the courier c has the unfinished task set $\mathcal{V} = \{o_p^1, o_p^2, \dots, o_p^n, o_d^1, o_d^2, \dots, o_d^n\}$, where o represents the order from customers, and o_p^i and o_d^i denote the pick-up task and the delivery task, respectively. We demonstrate the overall architecture of MRGRP in Figure 1, which consists of a multi-relational graph encoder and a reference-based courier route decoder. Specifically, we first construct a multi-relational graph based on these unfinished tasks, where nodes represent the tasks themselves and edges are defined based on their correlations. The encoder processing on multi-relational graphs transforms task features into highly expressive embeddings. After that, at each decoding step, the decoder recurrently predicts the courier's route selection by utilizing dynamic context information, reference solutions, and the learned task embeddings.

3.1 Multi-relational Graph Construction

Different from the existing method [34], which constructs a task graph containing only a single type of relation that integrates all spatial-temporal correlations between tasks, our approach represents tasks using a multi-relational graph at a more fine-grained level, which can benefit the learning of informative embeddings.

Edge definition. We define the relations between tasks as follows:

- **Pick-up then delivery.** It is the relation between the pick-up task o_p^u and the delivery task o_d^u of the same order o^u . The element A_{ij}^{pd} of its adjacency matrix A^{pd} can be defined as follows:

$$A_{ij}^{pd} = \begin{cases} 1, & \text{if } i = o_p^u \text{ and } j = o_d^u, \\ 0, & \text{others.} \end{cases} \quad (2)$$

- **Spatial proximity.** It indicates that the geographical distance between node $u, v \in \mathcal{V}$ is relatively close. The adjacency matrix for the spatial proximity relation can be denoted as A^{sc} , and its elements are defined as follows:

$$A_{ij}^{sc} = A_{ji}^{sc} = \begin{cases} 1, & \text{if } i, j \text{ are } k\text{-nearest spatial neighbors,} \\ 0, & \text{others,} \end{cases} \quad (3)$$

where k is a hyperparameter that can be tuned according to the cardinality of the task set \mathcal{V} .

- **Temporal proximity.** We denote the set of pick-up and delivery tasks as O_p and O_d , respectively. The relation indicates the promised pick-up (delivery) time between node $u, v \in O_p (O_d)$ is relatively close. The adjacency matrix of the temporal proximity

relation can be denoted as A^{pc} (A^{dc}), and its element A_{ij}^{pc} (A_{ij}^{dc}) is defined as follows:

$$A_{ij}^{pc} (A_{ij}^{dc}) = \begin{cases} 1, & \text{if } i, j \text{ are } k\text{-nearest temporal neighbors,} \\ 0 & \text{others,} \end{cases} \quad (4)$$

where k is a hyperparameter.

Node features. Given a task $u \in \mathcal{V}$, its feature comprises multi-domain information, as detailed in Table 1. The features of task u can be classified into categorical and numerical types, which are denoted as \mathbf{x}_u^c and \mathbf{x}_u^n , respectively. Please note that our framework is designed to be flexible, allowing the inclusion of additional task features as extra dimensions to further enhance performance.

Table 1: Multi-domain information of tasks.

Information	Description
Basic info	Deliver service type, deliver time type, etc.
Restaurant-related info	Earliest pick-up time, restaurant GPS coordinate, etc.
Customer-related info	Promised delivery time, customer GPS coordinate, etc.
Courier-related info	Distance to the courier location.
Area-related info	Area statistics (the count of tasks per rider, average speed, on-time rates, etc.).

Edge features. In the multi-relational graph, each edge describes a kind of aforementioned relation between its two ends. Given an edge $r_{uv} \in \mathcal{E}$, its feature is denoted as \mathbf{z}_{uv}^r , where r denotes the relation, and u, v represent two edge endpoints, respectively. According to the aforementioned edge definition, $r \in \mathcal{R} = \{1, 2, 3, 4\}$. Their semantics are spatially close, temporally close for pickup tasks, temporally close for delivery tasks, and pick-up then delivery.

3.2 MR-Graph Encoder

The multi-relational graph (MR-graph) encoder generates highly expressive embeddings for pick-up and delivery tasks, consisting of two key components: i) a node feature encoder and ii) a GraphFormer module [14, 25, 29]. Each component synergistically enhances the model’s capability to capture complex relationships and dependencies inherent in these tasks. Additionally, we integrate an encoding module for the auxiliary task of estimated time of arrival (ETA) prediction, which supports and enhances the primary task of route prediction.

Node feature encoder. Node features of tasks can be classified into categorical and numerical types. Instead of directly feeding the raw features to downstream modules, we first employ a DeepFM [7] framework for feature processing, where sparse categorical features are mapped to dense embeddings and high-order feature interactions are performed adaptively without specially designed feature engineering:

$$\mathbf{x}_u = \text{DeepFM}(\mathbf{x}_u^c, \mathbf{x}_u^n), \quad (5)$$

where $\mathbf{x}_u \in \mathcal{d}_v$ denotes the output processed feature of task u from DeepFM. After that, \mathbf{x}_u are then mapped to d_h -dimension latent embeddings, formulated as:

$$\bar{\mathbf{x}}_u = \sigma(\mathbf{W}_v \mathbf{x}_u + \mathbf{b}_v), \quad (6)$$

where $\bar{\mathbf{x}}_u$ constitutes the initial embeddings of task u and can be utilized by downstream modules. σ denotes the non-linear activation function, $\mathbf{W}_v \in \mathbb{R}^{d_v \times d_h}$ and $\mathbf{b}_v \in \mathbb{R}^{d_h}$ are trainable parameters.

GraphFormer module. In the context of the multi-relational graph, traditional graph convolutional networks (GCNs) lack the capabilities to model global dependencies between tasks due to their narrow receptive fields only for connected neighbors and have a risk of over-smoothing in deep convolutional layers. Although Transformers are alternatives to perceiving long-range global interactions, they cannot fully leverage the structural knowledge provided by the multi-relational graph. To address these limitations, we propose a novel GraphFormer module that effectively leverages the complementary strengths of GCNs and Transformers to encode task features and their correlations. It integrates specially designed multi-relational graph convolutional (MRGC) layers between Transformer layers. Specifically, denote the outputs from the node feature encoder as $\bar{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}| \times d_v}$, we input $\bar{\mathbf{X}}$ to the first Transformer layer, formulated as follows:

$$\mathbf{H}_T^{(1)} = \text{softmax}\left(\frac{\mathbf{Q}^{(1)} \mathbf{K}^{(1)T}}{\sqrt{d_e}}\right) \mathbf{V}^{(1)}, \quad (7)$$

where $\mathbf{Q}^{(1)} = \bar{\mathbf{X}} \mathbf{W}_q^{(1)}$, $\mathbf{K}^{(1)} = \bar{\mathbf{X}} \mathbf{W}_k^{(1)}$, $\mathbf{V}^{(1)} = \bar{\mathbf{X}} \mathbf{W}_v^{(1)}$, and $\mathbf{W}_q^{(1)} \in \mathbb{R}^{d_v \times d_e}$, $\mathbf{W}_k^{(1)} \in \mathbb{R}^{d_v \times d_e}$, $\mathbf{W}_v^{(1)} \in \mathbb{R}^{d_v \times d_e}$ are trainable parameters. To enhance the representation capabilities of the model, we extend it by incorporating multi-head attention, formulated as follows:

$$\mathbf{H}_T^{(1)} = \text{Concat}(\text{head}_1^{(1)}, \text{head}_2^{(1)}, \dots, \text{head}_m^{(1)}) \mathbf{W}_m^{(1)}, \quad (8)$$

where we omit the layer superscript for simplicity. Each of head_i is calculated from Equ. (7), and $\mathbf{W}_m^{(1)} \in \mathbb{R}^{md_e \times d_e}$ is a trainable parameter matrix. The output of the Transformer layer is subsequently fed to the MRGC layer, which leverages task dependencies provided in the multi-relational graph. The architecture of the MRGC layer is shown in Figure 2. Specifically, we first map edge features \mathbf{z}_{uv}^r (Section 3.1) into d_h -dimension latent embeddings, denoted as $\bar{\mathbf{z}}_{uv}^r$. A significant distinction between the MRGC layer and the conventional graph convolutional layer lies in the incorporation of relational information. Inspired by existing works [3, 21, 27, 31], we employ a composition function $\phi(\cdot)$ to integrate both node and relation embeddings, which can be formulated as:

$$\mathbf{e}_o = \phi(\mathbf{e}_s, \mathbf{e}_r) = \mathbf{e}_s \odot \mathbf{e}_r, \quad (9)$$

where $\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, \odot represents the element-wise product [38], \mathbf{e}_s , \mathbf{e}_r , and \mathbf{e}_o denotes the embedding of the subject, relation, and object, respectively. Therefore, the embeddings of all nodes after the first MRGC layer, $\mathbf{H}_G^{(1)}$, can be obtained by aggregating information from its neighboring nodes and relations, formulated as:

$$\mathbf{h}_{G,v}^{(1)} = \sigma\left(\sum_{(u,r,v) \in \mathcal{N}_v} \mathbf{W}_r^{(1)} \phi(\mathbf{h}_{T,u}^{(1)}, \bar{\mathbf{z}}_{uv}^r)\right), \quad (10)$$

where $r \in \mathcal{R}$ and \mathcal{N}_v is a set of neighbors of node v . $\mathbf{h}_{T,u}^{(1)}$ denotes the u -th row of $\mathbf{H}_T^{(1)}$. $\mathbf{W}_r^{(1)}$ corresponds to the trainable weights for relation type r . After updating node embeddings, edge embeddings can also be updated as follows:

$$\mathbf{q}_{uv}^r = \mathbf{W}_{rel} \bar{\mathbf{z}}_{uv}^r, \quad (11)$$

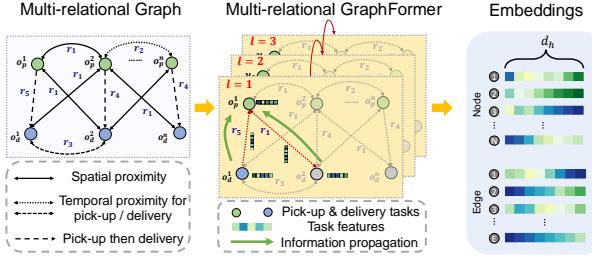


Figure 2: The schematic of MRGC layer.

where $\mathbf{W}_{rel} \in \mathbb{R}^{d_h \times d_h}$ aims to map all kinds of relations into the same embedding space, and \mathbf{q}_{uv}^r is the updated embedding of relation r . By stacking GraphFormer layers, we are capable of extracting fine-grained information from high-order neighbors and derive node embeddings recursively. The l -th GraphFormer layer can be represented as follows:

$$\mathbf{H}_T^{(l)} = \text{softmax}\left(\frac{\mathbf{Q}^{(l)} \mathbf{K}^{(l)T}}{\sqrt{d_e}}\right) \mathbf{V}^{(l)}, \quad (12)$$

$$\mathbf{H}_T^{(l)} = \text{Concat}(\text{head}_1^{(l)}, \text{head}_2^{(l)}, \dots, \text{head}_m^{(l)}) \mathbf{W}_m^{(l)}, \quad (13)$$

$$\mathbf{h}_{G,v}^{(l)} = \sigma\left(\sum_{(u,r,v) \in \mathcal{N}_u} \mathbf{W}_r^{(l)} \phi(\mathbf{h}_{T,v}^{(l)}, \mathbf{q}_{uv}^{r(l-1)}))\right), \quad (14)$$

where l denotes the number of layers, $\mathbf{Q}^{(l)} = \mathbf{H}_G^{(l-1)} \mathbf{W}_q^{(l)}$, $\mathbf{K}^{(l)} = \mathbf{H}_G^{(l-1)} \mathbf{W}_k^{(l)}$, and $\mathbf{V}^{(l)} = \mathbf{H}_G^{(l-1)} \mathbf{W}_v^{(l)}$. $\mathbf{H}_G^{(l)}$ represents the embeddings of all nodes from the l -th MRGC layer. Similarly, edge embeddings after l MRGC layers can be denoted as follows:

$$\mathbf{q}_{uv}^{r(l)} = \mathbf{W}_{rel}^{(l)} \mathbf{q}_{uv}^{r(l-1)}. \quad (15)$$

After propagating L GraphFormer layers, we stack an additional Transformer layer and obtain the final embeddings \mathbf{H} of tasks and \mathbf{Q}^G of relations.

3.3 Reference-based Route Decoder

Route decoder predicts the next finished task of the courier, with i) a masking mechanism to meet route constraints and avoid unreasonable predictions and ii) a reference-guided dynamic route decoding component for unfinished tasks.

Mask Mechanism. At each decoding step, we need to mask infeasible actions to meet route constraints and avoid unreasonable results. The mask mechanism includes the following masking rules:

- **Mask the tasks outputted at previous decoder steps.** The output task set before the prediction at decoding step z can be denoted as \mathcal{V}_z^P , where the order of tasks has been predicted and we should avoid selecting them repeatedly.
- **Mask the unfinished delivery tasks whose corresponding pick-up task has not been finished.** It represents the pick-up then delivery route constraint since the food cannot be delivered to the customer if it has not been picked up yet.
- **Mask the tasks that are not k -nearest spatial of the last outputted node.** Couriers typically tend to choose spatially closed tasks to finish next. If a task does not belong to the k -nearest spatial neighbors of the most recently outputted node, it is less probable to be the next task the courier wants to complete.

During the decoding process, the mask is updated at each step according to the above rules. The mask mechanism allows us to cut

off some unreasonable routes and shrink the search space, which contributes to our model's prediction performance.

Reference-based dynamic route decoding. For route prediction problems, feasible solution space usually grows exponentially with the number of tasks. Existing works have developed competitive models for route prediction and achieved relatively superior performance [5, 6, 24, 34, 36, 41, 42]. Therefore, integrating these solutions during route decoding may provide an effective reference and guide the prediction process. Although the selection of the reference solution is flexible, it should account for the time and storage complexity of the method used to produce the reference solution. In our implementation, we utilize a competitive heuristic method TSFH [44] (Section 4.1.2 and Appendix A.1) that operates without requiring training or learnable parameters. It is also deployed on the Meituan Turing Online Platform for route prediction. Furthermore, the route prediction problem is inherently dynamic, as it evolves with the courier's shifting location and the progression of time. This necessitates the development of a decoding strategy capable of adapting to constantly changing spatial and temporal conditions. Considering these contexts, we propose our reference-based dynamic task decoder. The whole decoding process can be formulated as follows:

$$p(\pi|s; \theta) = \prod_{z=1}^Z p(\pi_z|s, \pi_{1:z-1}; \theta), \quad (16)$$

$$= \prod_{z=1}^Z p(\pi_z|g(f, c; \theta_e), r_{1:z}, \pi_{1:z-1}; \theta), \quad (17)$$

where s is the given problem instance. θ_e denotes the trainable parameters. f, c denote the task features and the courier features, respectively. r is the reference route prediction solution. $g(f, c; \theta_e)$ denotes the multi-relational graph encoder. In our parameterization approach, we employ an LSTM module to encode the historically predicted route $\pi_{1:z-1}$. The hidden state at step $z-1$, denoted as \mathbf{m}_{z-1} , serves as the encoded representation of the historical route information. Similarly, we employ another LSTM module to embed the former z steps of the reference route $r_{1:z}$, and the hidden state at step z , denoted as \mathbf{r}_z , provides a comprehensive representation of this reference route segment.

Given the significance of distance and the remaining time until the promised arrival in influencing the courier's route decisions, we enrich task embeddings at each decoding step z by incorporating dynamic distance and time features, as formulated below:

$$\mathbf{H}^z = \text{Concat}(\mathbf{H}, \mathbf{d}^{z-1}, \mathbf{t}^{z-1}), \quad (18)$$

where \mathbf{H} denotes the final embeddings derived from the MR-graph encoder. The term \mathbf{d}^{z-1} represents the distance between each task and the task selected at the previous decoding step $z-1$. Additionally, \mathbf{t}^{z-1} signifies the remaining time until the promised arrival time for all tasks, explicitly defined as $\mathbf{t}^{z-1} = \mathbf{T} - \hat{\mathbf{t}}^{z-1}$, where \mathbf{T} denotes the promised arrival time of all tasks, and $\hat{\mathbf{t}}^{z-1}$ refers to the estimated time of arrival (ETA) of the task selected at decoding step $z-1$ (detailed in the following part). It is important to note that elements corresponding to masked tasks in \mathbf{d}^{z-1} and \mathbf{t}^{z-1} at each decoding step are set to infinity.

We subsequently derive the selection probability distribution of candidate tasks, considering the set of masked nodes at decoding

step z , denoted as \mathcal{V}_{mask}^z , based on the aforementioned masking rules:

$$u_v^z = \begin{cases} \sigma(\mathbf{h}_v^z \mathbf{W}_1)^T \sigma([\mathbf{m}_{z-1}, \mathbf{r}_z, r_{ID}^z, \mathbf{e}_{co}] \mathbf{W}_2) & \text{if } v \notin \mathcal{V}_{mask}^z \\ -\infty, & \text{otherwise,} \end{cases} \quad (19)$$

where \mathbf{h}_v^z is the v -th row of \mathbf{H}^z , i.e., the enriched embedding of task v at decoding step z . r_{ID}^z denotes the ID embedding of the route selection at decoding step z from the reference solution. \mathbf{e}_{co} is the embedding of couriers to enable personalized route prediction. $\mathbf{W}_1, \mathbf{W}_2$ are trainable parameters. $\sigma(\cdot)$ denotes the non-linear activation function GeLU [8]. Finally, the output probability of the candidate task can be formulated as follows:

$$p_v^z = p(\pi_z = v | s, \pi_{1:z-1}; \theta) = \text{softmax}(u_v^z). \quad (20)$$

Then, the decoder predicts the task to be finished at step z :

$$\pi_z = \text{argmax}_k p_k^z. \quad (21)$$

Auxiliary task of arrival time prediction. We also introduce an auxiliary task of arrival time prediction to augment the route prediction process by providing dynamic time features in Equ. (18). Specifically, we employ another LSTM module to model the temporal relations of tasks. At decoding step z , the input to the LSTM module is the concatenation of \mathbf{h}_{π_z} and $\mathbf{d}_{\pi_{z-1}:\pi_z}$, where \mathbf{h}_{π_z} denotes the π_z -th row of the embedding matrix \mathbf{H} from the MR-graph encoder. $\mathbf{d}_{\pi_{z-1}}^{\pi_z}$ indicates the distance between the previous selected task π_{z-1} and the current selected task π_z . After that, we employ a multilayer perceptron (MLP) to map the hidden embedding of LSTM at step z , denoted as $\mathbf{h}_{\text{eta}}^z$, to the prediction result \hat{t}^z .

3.4 Model Training and Prediction

Here we introduce the training and prediction process of MRGRP. **Training.** Since the decoder produces the selection probability distribution of each candidate task, which can be treated as the output of a multi-class classification problem. Hence, we utilize the cross-entropy (CE) loss for the route prediction task, which can be formulated as follows:

$$\mathcal{L}_R = -\frac{1}{S} \sum_{s \in \mathcal{S}} \sum_{v \in \pi} y_v \log(p(y_v | \theta)), \quad (22)$$

where \mathcal{S} is the set of samples, \mathcal{T} represents the set of sampling time steps, and y_v denotes the order of task v in the ground truth of the sample. $p(y_v | \cdot)$ is the predicted probability of task v .

For the auxiliary task of arrival time prediction, we introduce the quantile MAE loss [15], formulated as follows:

$$\mathcal{L}_{\text{ETA}} = \frac{1}{9} \sum_{\gamma=0.1}^{0.9} \sum_{z=1}^Z \mathbb{I}_{\hat{t}^z \geq t^z} (1 - \gamma) |t^z - \hat{t}^z| + \mathbb{I}_{\hat{t}^z < t^z} \gamma |t^z - \hat{t}^z|, \quad (23)$$

where t^z denotes the ground truth of the arrival time. Therefore, the final loss function can be formulated as:

$$\mathcal{L} = \mathcal{L}_R + \alpha \mathcal{L}_{\text{ETA}}, \quad (24)$$

where α is a hyperparameter controlling the relative weight. In our experiments, we set it as 0.1. We utilize teacher forcing to facilitate the training of our framework, where we use the actual routes as the input when generating the predicted task at each decoding step. **Prediction.** MRGRP takes all unfinished tasks as its input and predicts the entire pick-up and delivery routes recurrently. At each

decoding step, we use the predicted task generated by the model in the previous decoding step as the input, as shown in Equ. (17).

Time complexity analysis. We evaluate the training efficiency of our framework on a single NVIDIA RTX 2080Ti GPU. Each epoch takes approximately 15 minutes, and it takes 20 epochs (5 hours) at maximum for our model to converge among all datasets (see Section 4 for details). This training duration is acceptable for our industry application, as the trained model is deployed for long-term use without the need for frequent retraining.

4 Experiments

Here, we conduct experiments to show the performance of our proposed framework and answer the following research questions:

- **RQ1:** How does MRGRP perform compared with different kinds of route prediction models?
- **RQ2:** How do different designs of MRGRP affect the model performance?
- **RQ3:** How does MRGRP perform in terms of stability when predicting routes of varying lengths?

4.1 Experimental Settings

4.1.1 Datasets. We utilize large-scale real-world food delivery service data from Meituan¹, an influential online food delivery service provider in China, to show the performance of our approach. Our datasets consist of pick-up and delivery task records of couriers from a month in three different tier cities in China, including Beijing, Chengdu, and Guiyang, which allows us to analyze the robustness and adaptability of our method in both highly developed and less developed urban contexts. Moreover, such city-level evaluation framework also aligns with existing studies in the field [5, 34]. Each pick-up or delivery task record in the dataset contains various features of the task illustrated in Table 1 and of the couriers. We aim to predict the future food pick-up and delivery task routes of couriers. The detailed statistics are illustrated in Table 2.

Table 2: Statistics of datasets (ANT: Average number of tasks).

Type	City	ANUT	#Samples
Food P&D	Beijing	6	859461
	Chengdu	6	865641
	Guangzhou	5	868881

4.1.2 Baselines. We implement representative baseline methods of route prediction for performance comparison, which include rule-based methods, traditional machine learning methods, and state-of-the-art deep models.

- **Rule-based models.** i) *DisGreedy*, which selects the nearest accessible task from the courier's current location. ii) *TimeRank*, which selects the most urgent task. iii) *TSFH* [42], a heuristic route prediction method (Appendix A.1).
- **Machine learning models.** *Osquare* [42], which trains a Light-GBM [12] model to recurrently predict the next finished task at each step and generate the whole route of the courier.
- **Deep models.** i) *FDNet* [6], which employs an LSTM [9] and a Pointer Network [32] for route prediction in food delivery service. ii) *DeepRoute* [36], which utilizes a Transformer architecture to predict the route for all unfinished tasks. iii) *Graph2Route* [34], which constructs a spatial-temporal graph from tasks, encoding

¹<https://www.meituan.com>.

Table 3: Overall performance comparison of baselines w.r.t.all metrics. The best results are marked in bold, and we use underlines to denote the second-best results.

Dataset Model	Beijing						Chengdu						Guangzhou								
	KRC	LSD	ED	SR	SR@200	HR@1	ACC@3	KRC	LSD	ED	SR	SR@200	HR@1	ACC@3	KRC	LSD	ED	SR	SR@200	HR@1	ACC@3
TimeRank	0.833	1.895	1.211	0.35	0.407	0.604	0.412	0.827	2.135	1.136	0.347	0.399	0.591	0.408	0.835	1.457	1.108	0.375	0.428	0.624	0.434
DisGreedy	0.926	0.777	0.615	0.641	0.756	0.797	0.693	0.929	0.742	0.545	0.66	0.755	0.813	0.713	0.925	0.546	0.552	0.664	0.752	0.813	0.701
Osquare	0.918	0.912	0.718	0.582	0.675	0.748	0.647	0.915	0.914	0.67	0.591	0.672	0.751	0.659	0.915	0.597	0.641	0.61	0.679	0.771	0.663
TSFH	0.949	0.404	0.505	0.705	0.815	0.841	0.757	0.954	0.374	0.431	0.732	0.820	0.860	0.779	0.946	0.327	0.477	0.707	0.788	0.846	0.753
FDNet	0.922	0.901	0.712	0.589	0.681	0.756	0.659	0.924	0.902	0.658	0.596	0.677	0.759	0.664	0.926	0.588	0.629	0.619	0.692	0.78	0.679
DeepRoute	0.928	0.852	0.669	0.602	0.707	0.768	0.671	0.917	0.114	0.673	0.59	0.666	0.76	0.661	0.931	0.526	0.577	0.636	0.701	0.789	0.685
JGRM	0.937	0.610	0.550	0.676	0.782	0.839	0.736	0.938	0.630	0.510	0.684	0.771	0.841	0.741	0.944	0.406	0.490	0.697	0.776	0.850	0.746
Graph2Route	0.953	0.344	0.478	0.721	0.824	0.852	0.769	0.958	0.325	0.407	0.751	0.831	0.877	0.801	0.952	0.282	0.444	0.743	0.815	0.87	0.787
DRL4Route	0.936	0.608	0.544	0.674	0.785	0.84	0.734	0.935	0.636	0.515	0.682	0.769	0.838	0.739	0.947	0.402	0.486	0.698	0.778	0.852	0.749
ILRoute	0.939	0.612	0.554	0.679	0.78	0.837	0.738	0.94	0.623	0.508	0.686	0.774	0.844	0.743	0.94	0.41	0.498	0.696	0.774	0.848	0.745
Ours	0.962	0.276	0.394	0.764	0.856	0.888	0.814	0.965	0.259	0.338	0.785	0.86	0.9	0.832	0.959	0.232	0.368	0.772	0.839	0.892	0.814
Improvement	0.94	19.77%	17.57%	5.96%	3.88%	4.23%	5.85%	0.73%	20.31%	16.95%	4.53%	3.49%	2.62%	3.87%	0.74%	17.73%	17.12%	3.90%	2.94%	2.53%	3.43%

their correlations. iv) *JGRM* [23], which learns route embeddings based on self-supervised methods. v) *DRL4Route* [24], which leverages actor-critic methods to optimize pick-up and delivery routes in dynamic environments. vi) *ILRoute* [5], which models the decision-making process of couriers with imitation learning. We perform a grid search of hyperparameters of deep learning methods with the validation set to ensure that they can achieve optimal performance for a fair comparison. We conduct each experiment at least five times and report the average results.

4.1.3 Evaluation metrics. We denote the predicted routes and the ground truths as $\hat{\pi} = [\hat{\pi}_1, \dots, \hat{\pi}_n]$ and $\pi = [\pi_1, \dots, \pi_m]$, respectively, where there could be $m \leq n$ as well as $\text{set}(\pi) \subseteq \text{set}(\hat{\pi})$. We define $S_\pi(i)$ and $S_{\hat{\pi}}(i)$ as the location of the i -th task in the ground truth and the predicted route, respectively. We utilize widely accepted metrics to evaluate performances of methods from both local and global perspectives, which include Kendall rank correlation (KRC) [13], location square deviation (LSD), edit distance (ED) [26], same rate@k (SR@k), Hit-Rate@k (HR@k), Accuracy@k (ACC@k). Definition details are demonstrated in Appendix A.2.

4.2 Overall Performance (RQ1)

We show the overall performances of models *w.r.t.* metrics in global and local perspectives in Table 3, respectively. From the experimental results, we have the following observations:

- **Our framework effectively achieves the best route prediction performance via MRGRP.** The evaluation results on datasets of three cities demonstrate that our proposed MRGRP can steadily achieve promising performance improvement by a large margin compared with all other baselines. For example, on Beijing dataset, MRGRP improves the performance of the state-of-the-art baseline (Graph2Route) by 19.77% at LSD, 17.57% at ED, and by 5.96% at SR, 4.23% at HR@1. It also consistently achieves the best route prediction results on the other two datasets. Considering that our model can be deployed to serve millions of couriers in Meituan, a slight increase in the offline experiment metrics can have a significant performance improvement in online scenarios.
- **Explicitly integrating relations between tasks facilitates accurate route prediction.** Besides the best performance of MRGRP, we find that Graph2Route [34], which constructs an ST graph from tasks, performs better than other baselines. Since couriers consider multiple factors during route planning, encoding complex correlations between tasks is beneficial to models.
- **Distance is an important feature in route prediction tasks.** We find that DisGreedy, a simple strategy to select the nearest tasks in each step, exhibits competitive performances and exceeds

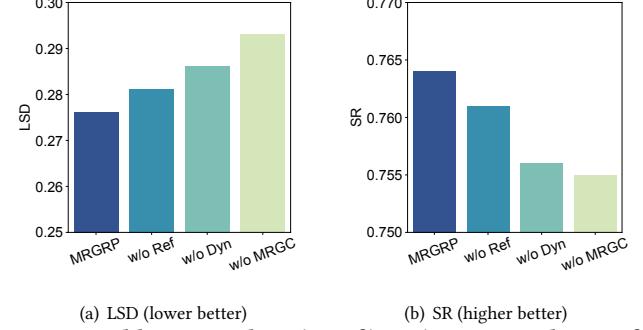


Figure 3: Ablation study. w/o Ref/Dyn/MRGC: without reference route solutions/dynamics features during decoding/MRGC layers.

some deep models such as FDNet [6] and DeepRoute [36]. It underscores the significance of distance as a critical feature of tasks. Regarding distances as ordinary features without explicit modeling may result in their information being conflated with other features, thereby comprising the route prediction performance.

4.3 Ablation Study (RQ2)

We conduct ablation studies to verify the contributions of designed components in MRGRP. Specifically, we remove the main components of MRGRP in turn and report the experimental results on Beijing dataset in Figure 3.

- **Without MRGC layers.** We exclude MRGC layers from the encoder, retaining only Transformer layers. The results indicate a 0.017 increase in LSD and a 0.010 decrease in SR, underscoring the critical importance of the constructed multi-relational graph. It demonstrates that leveraging explicit, structured knowledge of task interdependencies significantly benefits the learning of their embeddings and complex correlations.
- **Without dynamics features during decoding.** We remove the dynamic distance and time features in Equ. (18) at each decoding step and observe a 0.01 increase in LSD and a 0.008 decrease in SR. It underscores the necessity of carefully considering the inherent dynamism of problem conditions in route prediction.
- **Without reference route solutions.** We remove the reference task selection at each decoding step. Under this configuration, the LSD of the model increases by 0.005 while its SR decreases by 0.003. These findings highlight the essential role of reference route solutions in guiding the discovery of more optimal routes and effectively narrowing the solution space.

4.4 Robustness against Route Lengths (RQ3)

In this section, we investigate the robustness of MRGRP with respect to route lengths, defined by the number of unfinished tasks

in each data sample. Samples with 5 or more unfinished tasks are categorized as long-route samples, while those with fewer than 5 unfinished tasks are categorized as short-route samples. We train and evaluate MRGRP, along with other baseline models, on both categories separately to assess their respective performances. Due to the space limit, we demonstrate LSD, ED, SR, and ACC@3 of models on Beijing dataset in Table 4. The results indicate that MRGRP consistently outperforms all baseline methods on both long-route and short-route samples. Specifically, it achieves improvements in LSD and ACC@3 by 6.69% and 7.49%, respectively, on long-route samples and by 24.53% and 4.48%, respectively, on short-route samples. The superior performance can be attributed to the ability of MRGRP to effectively capture and leverage the complex task correlations provided by the multi-relational graph.

Table 4: Performances on long and short-route samples.

Model	Long routes				Short routes			
	LSD	ED	SR	ACC@3	LSD	ED	SR	ACC@3
TimeRank	3.484	1.744	0.204	0.273	0.572	0.793	0.513	0.52
DisGreedy	1.387	0.885	0.541	0.632	0.301	0.404	0.753	0.74
Osquare	1.482	0.951	0.495	0.586	0.397	0.524	0.691	0.725
TSFH	0.725	0.778	0.589	0.667	0.153	0.291	0.835	0.828
FDNet	1.785	1.18	0.413	0.505	0.235	0.371	0.777	0.772
DeepRoute	1.717	1.023	0.437	0.521	0.384	0.433	0.698	0.695
JGMR	1.092	0.855	0.552	0.638	0.192	0.313	0.802	0.797
Graph2Route	0.523	0.659	0.642	0.694	0.159	0.295	0.835	0.826
DRL4Route	1.099	0.862	0.55	0.641	0.198	0.325	0.791	0.776
ILRoute	1.081	0.848	0.557	0.647	0.185	0.31	0.813	0.806
MRGRP	0.488	0.618	0.668	0.746	0.12	0.231	0.864	0.863
Improvement	6.69%	6.22%	4.05%	7.49%	24.53%	21.69%	3.47%	4.48%

5 Online Deployed Evaluation

We have incorporated and deployed MRGRP to the Meituan Turing Platform for more than a month, which operates one of the largest food delivery systems in China and processes millions of daily orders to predict the routes of couriers. We train MRGRP with 10 million real courier routes randomly sampled from country-level data and test its performance on randomly sampled 50k routes. We also sample test data according to their lengths and evaluate the performance of MRGRP. The results, as presented in Table 5, demonstrate the superiority of our approach compared to the already deployed TSFH [44] on the platform. We find that MRGRP achieves significant improvements on routes with random lengths across various metrics, yielding 31.85% and 7.46% improvement in LSD and SR, respectively. MRGRP exhibits an 11.52% improvement in SR for datasets with long routes and a 4.75% improvement for short routes. These results underscore the effectiveness of MRGRP in route prediction across diverse real-world scenarios. When utilizing our predicted routes for arrival time estimation, we observe a decrease in its MAE by 14.61%, 16.07%, and 12.98% on random, long, and short routes, respectively. These improvements significantly benefit delivery efficiency, user satisfaction and retention of the instant food delivery service provided by Meituan.

Table 5: Results of online deployed evaluation.

Model	Random routes		Long routes		Short routes	
	LSD	SR	LSD	SR	LSD	SR
TSFH	0.416	0.762	0.647	0.692	0.154	0.839
MRGRP	0.284	0.819	0.419	0.772	0.134	0.879
Improvement	31.85%	7.46%	35.22%	11.52%	13.20%	4.75%

6 Related Works

Related works are classified into two groups: next location prediction and route prediction.

Next Location Prediction. This task is to predict which locations an individual will visit in future time slots. Many researchers proposed deep learning-based approaches to address this task [1, 4, 10, 16, 19, 28, 30, 37, 39, 40, 43]. For example, by incorporating spatial-temporal influence, HST-LSTM [16] utilizes LSTM to forecast the movement of users in the upcoming minutes or hours. CTLE [20] proposed a location embedding model that considers both contextual and temporal information in trajectories. TCSA-Net [30] developed a temporal-context-based self-attention network to capture both long-term and short-term preferences of user movements. A common of these works is to infer the possible location from the whole location set in the data. Different from them, our work focuses on selecting the location of the pick-up or delivery task from a given set containing the locations of unfinished tasks. Meanwhile, since the pick-up and delivery tasks in food delivery services have strict spatial and temporal constraints, it is hard to use the above-mentioned methods in our work directly.

Route Prediction. Route prediction is previously formulated as a path-finding problem and addressed by minimizing the search space with heuristic algorithms [22, 33]. Recently, more researchers have studied the problem of route prediction with learning approaches in this field [5, 6, 24, 34–36, 42]. For example, [42] designed a machine-learning model to extract the features of decision-making psychology to predict couriers' future routes. DeepRoute [36] proposed a novel deep model to predict couriers' routes, where a transformer encoder is utilized to capture the spatio-temporal dependencies between unpick-up packages, and an attention-based decoder is adopted to infer the pick-up order of the unfinished tasks recurrently. Graph2Route [34] designed a spatio-temporal graph neural network to explore the spatio-temporal correlations between different tasks and utilized the graph structure and couriers' profile to decode future routes. ILRoute [5] utilized a graph-based imitation learning framework that models the decision-making process of food delivery couriers. However, these works cannot fully explore how factors related to pickup and delivery decision-making affect courier behavior. Unlike them, we propose to build a multi-relational graph to consider these factors and employ the effective reference solution and dynamical environment contexts to improve the accuracy of route prediction.

7 Conclusion

We propose a powerful approach, MRGRP, to predict the delivery routes. It utilizes a specifically designed MR-graph encoder to capture intricate correlations of tasks. The decoder leverages existing route solutions as references and couriers' personalized features during predictions while narrowing the solution space. Additionally, it integrates dynamic distance and time contexts, considering the inherent dynamism of delivery conditions. When deployed on the Meituan Turing Platform, it significantly improves the currently used algorithm, substantially enhancing delivery efficiency, user satisfaction, and retention.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under 62272260, 62476152, U24B20180.

References

- [1] Meng Chen, Yixuan Zuo, Xiaoyi Jia, Yang Liu, Xiaohui Yu, and Kai Zheng. 2020. CEM: A convolutional embedding model for predicting next locations. *IEEE TITS* 22, 6 (2020), 3349–3358.
- [2] Arthur Cruz de Araujo and Ali Etemad. 2021. End-to-End Prediction of Parcel Delivery Time With Deep Learning for Smart-City Applications. *IEEE Internet Things J.* 8, 23 (2021), 17043–17056.
- [3] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI (2018)*, Vol. 32.
- [4] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW (2018)*, 1459–1468.
- [5] Tao Feng, Huan Yan, Huandong Wang, Wenzhen Huang, Yuyang Han, Hongsen Liao, Jinghua Hao, and Yong Li. 2023. ILRoute: A Graph-based Imitation Learning Method to Unveil Riders' Routing Strategies in Food Delivery Service. In *SIGKDD (2023)*, 4024–4034.
- [6] Chengliang Gao, Fan Zhang, Guanqun Wu, Qiwan Hu, Qiang Ru, Jinghua Hao, Renqin He, and Zhihao Sun. 2021. A Deep Learning Method for Route and Time Prediction in Food Delivery Service. In *SIGKDD (2021)*, 2879–2889.
- [7] Huiwen Guo, Ruiming Tang, Yuming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).
- [8] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).
- [9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [10] Ye Hong, Henry Martin, and Martin Raubal. 2022. How do you go where? Improving next location prediction by learning travel mode information using transformers. In *SIGSPATIAL (2022)*, 1–10.
- [11] Wenjie Huang, Zhiwei Zhao, Xiaolong An, Geyong Min, and Jingjing Li. 2020. Dynamic scheduling for urban instant delivery with strict deadlines. In *IEEE ICC (2020)*, 1–6.
- [12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS (2017)* 30.
- [13] Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.
- [14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (2017)*. OpenReview.net.
- [15] Roger Koenker. 2005. *Quantile regression*. Vol. 38. Cambridge university press.
- [16] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction. In *IJCAI (2018)*. AAAI Press, 2341–2347.
- [17] Yu-Lin Lan, Fagui Liu, Wing WY Ng, Jun Zhang, and Mengke Gui. 2020. Decomposition based multi-objective variable neighborhood descent algorithm for logistics dispatching. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 5 (2020), 826–839.
- [18] Yiyao Li and William Phillips. 2018. Learning from route plan deviation in last-mile delivery. (2018).
- [19] Yixin Li, Yu Zheng, and Qiang Yang. 2020. Cooperative multi-agent reinforcement learning in express system. In *CIKM (2020)*, 805–814.
- [20] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2021. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *AAAI (2021)*, Vol. 35. 4241–4248.
- [21] Chang Liu, Chen Gao, Yuan Yuan, Chen Bai, Lingrui Luo, Xiaoyi Du, Xinlei Shi, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Modeling Persuasion Factor of User Decision for Recommendation. In *SIGKDD (2022)*, 3366–3376.
- [22] Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M Ni. 2013. Finding time period-based most frequent path in big trajectory data. In *SIGMOD (2013)*, 713–724.
- [23] Zhipeng Ma, Zheyuan Tu, Xinhai Chen, Yan Zhang, Deguo Xia, Guyue Zhou, Yilun Chen, Yu Zheng, and Jianguo Gong. 2024. More Than Routing: Joint GPS and Route Modeling for Refine Trajectory Representation Learning. In *WWW (2024)*, 3064–3075.
- [24] Xiaowei Mao, Haomin Wen, Hengrui Zhang, Huaiyu Wan, Lixia Wu, Jianbin Zheng, Haoyuan Hu, and Youfang Lin. 2023. Dr4route: A deep reinforcement learning framework for pick-up and delivery route prediction. In *SIGKDD (2023)*, 4628–4637.
- [25] Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *EMNLP*. Association for Computational Linguistics, 1506–1515.
- [26] John Nerbonne, Wilbert Heeringa, and Peter Kleiweg. 1999. Edit distance and dialect proximity. *Time Warps, String Edits and Macromolecules: The theory and practice of sequence comparison* 15 (1999).
- [27] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2015), 11–33.
- [28] Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo Ungerer. 2005. Next location prediction within a smart office building. *Cognitive Science Research Paper-University of Sussex CSRP* 577 (2005), 69.
- [29] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC (Lecture Notes in Computer Science, Vol. 10843)*. Springer, 593–607.
- [30] Guiming Sun, Heng Qi, Yanming Shen, and Baocai Yin. 2022. Tcsa-net: a temporal-context-based self-attention network for next location prediction. *IEEE TITS* 23, 11 (2022), 20735–20745.
- [31] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082* (2019).
- [32] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).
- [33] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In *SIGKDD (2012)*, 195–203.
- [34] Haomin Wen, Youfang Lin, Xiaowei Mao, Fan Wu, Yiji Zhao, Haochen Wang, Jianbin Zheng, Lixia Wu, Haoyuan Hu, and Huaiyu Wan. 2022. Graph2Route: A Dynamic Spatial-Temporal Graph Neural Network for Pick-up and Delivery Route Prediction. In *KDD (2022)*, 4143–4152.
- [35] Haomin Wen, Youfang Lin, Huaiyu Wan, Shengnan Guo, Fan Wu, Lixia Wu, Chao Song, and Yinghui Xu. 2022. DeepRoute+: Modeling Couriers' Spatial-temporal Behaviors and Decision Preferences for Package Pick-up Route Prediction. *ACM TIST* 13, 2 (2022), 1–23.
- [36] Haomin Wen, Youfang Lin, Fan Wu, Huaiyu Wan, Shengnan Guo, Lixia Wu, Chao Song, and Yinghui Xu. 2021. Package pick-up route prediction via modeling couriers' spatial-temporal behaviors. In *ICDE (2021)*. IEEE, 2141–2146.
- [37] Huan Yan and Yong Li. 2025. A Survey of Generative AI for Intelligent Transportation Systems: Road Transportation Perspective. *Comput. Surveys* (2025).
- [38] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR (2015)*.
- [39] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location prediction over sparse user mobility traces using rnns. In *IJCAI (2020)*, 2184–2190.
- [40] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. Serm: A recurrent model for next location prediction in semantic trajectories. In *CIKM (2017)*, 2411–2414.
- [41] Jinhui Yi, Huan Yan, Haotian Wang, Jian Yuan, and Yong Li. 2024. Learning to Estimate Package Delivery Time in Mixed Imbalanced Delivery and Pickup Logistics Services. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 432–443.
- [42] Yan Zhang, Yunhai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. 2019. Route prediction for instant delivery. *IMWUT* (2019) 3, 3 (2019), 1–25.
- [43] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE TKDE* 34, 5 (2020), 2512–2524.
- [44] H Zheng, S Wang, Y Cha, F Guo, J Hao, R He, and Z Sun. 2019. A two-stage fast heuristic for food delivery route planning problem. In *Informs annual meeting, Seattle, Washington, USA*.

A Appendix

In this appendix, we introduce additional details of the baseline deployed algorithms, evaluation metrics, and experimental results.

A.1 TSFH Algorithm

The heuristic algorithm TSFH [44] operates through a two-step process involving initialization and local search strategies. This algorithm optimizes route scheduling by considering both the distance between tasks and their estimated time of arrival (ETA). Each order consists of a pick-up task and a delivery task. During the initialization phase, the algorithm sorts all orders according to their ETA and plans the first order according to the pick-up then delivery constraint. For subsequent orders, the algorithm greedily inserts pick-up and delivery tasks into the route, aiming to minimize delays and route length. To speed up the initialization process, the algorithm utilizes geographic information to cluster pick-up and delivery tasks, considering that groups of restaurants and customers are often geographically proximate. This clustering method

effectively reduces sub-optimal insertion attempts, ensuring that each task is primarily inserted within its respective cluster. After the initialization stage, the algorithm applies a local search to further refine the solution by exploring its neighborhood. During this phase, the algorithm iteratively identifies delivery tasks with significant time margins. Specifically, tasks with an ETA more than five minutes earlier than the promised delivery time are adjusted backward to optimize the schedule. Conversely, tasks with delays exceeding five minutes are moved forward to mitigate lateness. The solution is continuously updated whenever a better route is discovered through the local search.

A.2 Details of Evaluation Metrics

KRC: Kendall Rank Correlation (KRC) [13] is a statistical measurement to the ordinal association between two sequences. For a task pair (u, v) , it is regarded as a concordant pair when both $S_{\hat{\pi}}(u) > S_{\hat{\pi}}(v)$ and $S_{\pi}(u) > S_{\pi}(v)$, or $S_{\hat{\pi}}(u) < S_{\hat{\pi}}(v)$ and $S_{\pi}(u) < S_{\pi}(v)$. Otherwise, it is regarded as a discordant pair. As mentioned before, since the tasks in the predicted route may not exist in the ground truth, we categorize them into two sets, consisting of i) tasks in the ground truth ($\mathcal{V}_{in} = \{\hat{\pi}_i | \hat{\pi}_i \in \pi\}$) and ii) tasks not in the ground truth ($\mathcal{V}_{not} = \{\hat{\pi}_i | \hat{\pi}_i \notin \pi\}$). The order of tasks in \mathcal{V}_{in} are easy to obtain, and we can also know that the order of tasks in \mathcal{V}_{in} are ahead of that in \mathcal{V}_{out} . Consistent with the existing work [34], we compare task pairs $\{(u, v) | u, v \in \mathcal{V}_{in}, u \neq v\} \cup \{(u, v) | u \in \mathcal{V}_{in}, v \in \mathcal{V}_{out}\}$. The KRC is defined as follows:

$$KRC = \frac{N_c - N_d}{N_c + N_d}, \quad (25)$$

where N_c and N_d denotes the number of concordant and discordant task pairs, respectively.

LSD: The Location Square Deviation (LSD) is used to measure the degree that the predicted route deviates from the ground truth, which can be formulated as:

$$LSD = \frac{1}{m} \sum_{i=1}^m (S_{\pi}(\pi_i) - S_{\hat{\pi}}(\pi_i))^2. \quad (26)$$

ED: Edit Distance [26] is the metric to measure the dissimilarity of sequences via counting the minimum required steps to transform one sequence to another.

Same rate@k (SR@k): It quantifies the relaxed consistency rate of the predicted route and the ground truth. In particular, we measure it using the ratio of the common prefix length of two sequences to the length of the ground truth. k represents a specified distance. If the distance between the predicted node and the node in the ground truth is less than k meters, we consider the model to provide accurate prediction results at this step and count it into the common prefix. The metrics are designed to relax the evaluation of prediction results and are reasonable in real-world scenarios. For example, the courier can pick up two tasks at the same merchant; it is meaningless to evaluate the order of two tasks strictly. In our experiments, we use SR@1 and SR@200 as the evaluation metrics.

HR@k: Hit-Rate@k is used to measure the similarity between top- k items of two sequences, which can be formulated as follows:

$$HR@k = \frac{|\hat{\pi}_{[1:k]} \cap \pi_{[1:k]}|}{k}. \quad (27)$$

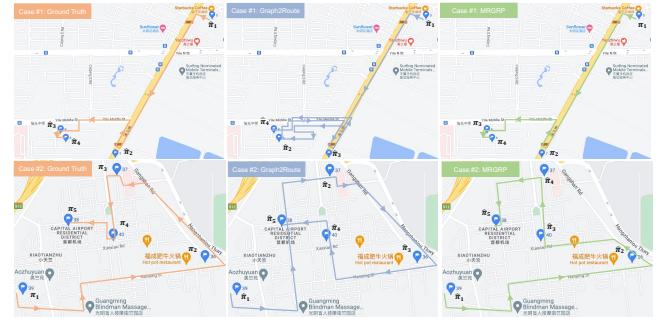


Figure 4: Case study. Orange, blue, and green lines correspond to real routes, routes predicted by Graph2Route, and MRGRP. ACC@k: ACC@k is used to calculate the local similarity of two sequences. It measures whether the first k tasks of the predicted route are exactly the same as those of the ground truth, which can be formulated as follows:

$$ACC@k = \prod_{i=0}^k \mathbb{I}(\hat{\pi}_i, \pi_i), \quad (28)$$

where $\mathbb{I}(\hat{\pi}_i, \pi_i) = 1$ when $\hat{\pi}_i = \pi_i$, otherwise $\mathbb{I}(\hat{\pi}_i, \pi_i) = 0$.

A.3 Further Experimental Results

A.3.1 Case Study. To provide an intuitive analysis of MRGRP, we use empirical route prediction cases to illustrate its performance and advantages over the state-of-the-art model (Graph2Route), which are shown in Figure 4. For each case, we illustrate the real routes of a courier, routes predicted by Graph2Route and MRGRP.

- Case #1:** We find that MRGRP predicts a more reasonable and accurate route compared to the baseline Graph2Route, which is also consistent with the ground truth. There are four pick-up and delivery tasks for two orders denoted by 5 ~ 8 on the map, where 5 and 7 are pick-up tasks, and 6 and 8 are corresponding delivery tasks. The prediction of MRGRP is identical to the actual route of the courier and is an efficient solution. However, the route predicted by Graph2Route is unreasonable; the courier needs to enter and depart the same residential area twice, and the length of the route is much longer than the actual route.
- Case #2:** In this case, there are five pick-up and delivery tasks for three orders denoted by 36 ~ 40 on the map, where 37 and 39 are pick-up tasks and 36, 38, and 39 are delivery tasks. Despite the fact that both MRGRP and Graph2Route fail to accurately predict the entire path, we find that the predicted route of MRGRP is also reasonable, and the route length does not differ significantly from the ground truth, which is less likely to cause serious harm to downstream tasks such as arrival time prediction and order dispatching. However, the route predicted by Graph2Route deviates much from the ground truth. For example, it predicts the task 37 in the second step, which is the farthest from the previous predicted task 39. Therefore, the prediction results are not reliable and cannot contribute much to the other tasks.

A.3.2 Hyperparameter Study. We performed comprehensive hyperparameter tuning to identify the optimal configuration for model performance. Our study revealed that setting the number of MRGC layers to 3, the task embedding size to 256, and the relative weight of the loss functions to 0.1 yielded the best results. Due to space constraints, detailed results are provided in our GitHub repository <https://github.com/tsinghua-fib-lab/MGRoute>.