Talking Trails: LLM-Enhanced Spatiotemporal Trajectory Modeling for E-Bike Delivery Route Planning

Zhao Li[1,2]*, **Mingwu Liu**[3]*, **Xu-Hua Yang**[3], **Haipeng Dai**[4], **Ji Zhang**[5]†, **Yangbohan Jiao**[1]†
[1]Hangzhou Yugu Technology Co., Ltd., Hangzhou, China
[2]Zhejiang Lab, Hangzhou, China
[3]Zhejiang University of Technology, Hangzhou, China
[4]Nanjing University, Nanjing, China
[5]University of Southern Queensland, Toowoomba, Australia
lzjoey@gmail.com, mw_liu@zjut.edu.cn, xhyang@zjut.edu.cn, haipengdai@nju.edu.cn, zhangji77@gmail.com,
yangbohanjiao@gmail.com

Abstract

Electric bicycles (e-bikes) have become the dominant mode of transportation in China's urban instant delivery industry. However, many riders lack the experience to navigate complex traffic networks and diverse road conditions, leading to reduced delivery efficiency. To address this issue, we present Talking Trails, an e-bike delivery route planning system built upon an LLM-enhanced spatiotemporal trajectory model. Trained on millions of real-world delivery trajectories, fused with spatiotemporal and semantic data information, the model achieves a top-5 rider displacement prediction accuracy of 95% and a route optimization rate of 82.1%. In practice, we augment the core planner with an LLM-driven semantic layer that translates high-level user intent into executable tasks, then pair it with a battery-swap module that continuously validates route feasibility so the vehicle never runs out of charge mid-mission. Currently serving tens of thousands of riders, the system is projected to reduce average delivery mileage by 17% and lower annual carbon emissions by 3978 tons. Overall, Talking Trails significantly improves delivery efficiency, offering a scalable and sustainable solution for instant delivery operations.

Introduction

In China, electric bicycles (e-bikes) have become the primary mode of transportation for high-intensity and high-frequency urban instant delivery services. The delivery industry now handles over 10 billion orders annually, with continued growth expected in the coming years. For delivery riders, performance is closely tied to efficiency: faster and more punctual deliveries directly contribute to higher earnings. However, the complexity of urban traffic conditions, such as sudden road closures, heavy congestion, and varying pedestrian activity, creates significant challenges for riders. In this context, route planning and navigation experience are crucial to reducing delivery times and ensuring timely deliveries. However, the conventional route planning method does not provide adequate solutions for riders. Most of the methods, which rely largely on static road networks, do not capture the personalized decisions riders make in real time based on experience, such as opting for quieter side streets over more congested main roads. As a result, the lack of appropriate tools leaves independent riders under prolonged delivery pressure.

Currently, route planning methods can be broadly categorized into two paradigms: rule-based algorithms and deep learning-based approaches. Rule-based methods, such as Dijkstra's (Ding, Yu, and Qin 2008) and A* (Lu, Chen, and Tseng 2016; Boczka et al. 2025), along with their dynamic variants including D (Stentz 1994) and D* Lite (Koenig and Likhachev 2002), model road networks as static graphs and compute shortest paths (Kanoulas et al. 2006). While effective in deterministic environments, these methods find it challenging to capture the dynamic and personalized decision-making of e-bike riders, such as avoiding congestion, favoring quieter routes, or choosing rider-friendly roads. This limitation makes them less adaptable to the complex and fast-changing conditions of real-world instant delivery scenarios.

In contrast, deep learning-based methods have emerged as a more adaptive alternative by learning from historical trajectory data (Chen et al. 2024). These models can capture complex spatiotemporal patterns and incorporate diverse contextual factors, including traffic conditions, road topology, and rider behavior (Salzmann et al. 2020; Li et al. 2021; Ma et al. 2021). Spatiotemporal modeling, in particular, has shown promise in improving route prediction accuracy by jointly modeling spatial dependencies and temporal dynamics (Alhussen and Ansari 2024). For instance, CNNs and RNNs have been successfully applied to trajectory prediction and route optimization in urban mobility (Sadid and Antoniou 2024; Hsu et al. 2023), while attention mechanisms enable dynamic weighting of time-varying factors such as traffic fluctuations and time-of-day effects (Zhang, Zhu, and Ma 2023). This progress in spatiotemporal modeling has motivated recent research to integrate finer-grained personalization and micro-mobility-specific constraints into route planning frameworks. Ye et al. (Ye et al. 2025) incorporated user perception and scene semantics for shared scooter routing, Nourmohammadi et al. (Nourmohammadi, Rey, and Saberi 2025) optimized

---

*These authors contributed equally.

†Corresponding authors

e-cargo routes under cycling fatigue constraints, and Amin et al. (Amin, Amin, and Cho 2023) leveraged machine learning to predict battery state-of-charge for energy-aware navigation. Lee et al. (Lee, Han, and Song 2023) further demonstrated that traditional graph representations are inadequate for modeling fine-grained micro-mobility behaviors. Despite these advances, most existing deep learning models are adapted from general vehicle routing frameworks and are not well suited to capturing the unique dynamics of e-bikes, such as their compact size, high maneuverability in mixed-traffic environments, and distinct speed and acceleration profiles. As a result, these models remain suboptimal for high-frequency, time-critical instant delivery tasks, where rider behavior and micro-mobility-specific constraints significantly influence route choices.

A recent and promising direction lies in enhancing spatiotemporal models with LLMs, which have demonstrated remarkable capabilities in semantic understanding and zero-shot generalization (Li et al. 2024). Methods such as PromptCast (Xue and Salim 2023) and LLMTime (Gruver et al. 2023) convert time series data into textual prompts and utilize LLMs for multi-task forecasting, achieving strong performance in transfer and zero-shot settings. Sun et al. (Sun et al. 2024) explore instance-aligned time series embeddings and fine-tuning strategies to improve cross-task adaptability, while AutoTimes (Liu et al. 2024) redefines LLMs as autoregressive time series predictors capable of generating arbitrarily long future sequences. By integrating textual semantics, these models can better interpret user intent and environmental context (Wu and Ling 2024; Panagoulias, Virvou, and Tsihrintzis 2024), leading to more human-aligned predictions. This capability is especially valuable for e-bike delivery, where route decisions depend not only on spatial and temporal dynamics but also on rich semantic cues, including rider preferences, descriptions of road conditions, and delivery-specific requirements.

In this paper, we propose Talking Trails, a novel e-bike instant delivery route planning method based on an LLM-enhanced spatiotemporal trajectory model. The model is trained on large-scale real-world trajectory data from delivery riders, incorporating a combination of semantic and trajectory-based features, including user information, temporal sequences, position sequences, battery power sequences, and semantic descriptions of route segments. This enables the model to learn from the extensive route planning experience and preference of delivery riders. We integrate the model with heuristic search techniques to generate faster and more reliable routes. Experimental results show that Talking Trails achieves a reachability rate of 91.5%, with 82.1% of the generated routes outperforming those chosen by real delivery riders. To ensure practical deployment, Talking Trails integrates LLM-based task augmentation techniques to support end-to-end semantic service flows, and incorporates battery energy data for battery swapping decisions, helping alleviate riders' range anxiety. We make three major contributions in this paper:

- We propose an LLM-enhanced spatiotemporal trajectory framework that integrates multi-dimensional trajectory features and descriptive textual information at both the data and model levels, substantially improving the model's representational and learning capabilities.
- We design a spatiotemporal trajectory model coupled with a heuristic path generation mechanism, which significantly increases the efficiency of route planning in the instant delivery domain.
- We integrate our path generation approach with LLM and real-world geographic data to construct a semantic-to-semantic interactive navigation system. This system enables a closed-loop process from user semantic input to route recommendation, providing users with intelligent, dynamic, and precise personalized delivery navigation services.

Talking Trails Framework

Talking Trails performs route planning via an iterative two-stage process: trajectory point prediction and point-by-point inference, as illustrated in Figure 1(a) and (b), respectively. In the first stage, a semantic-enhanced spatiotemporal trajectory model predicts a set of candidate next locations by integrating historical movement patterns with semantic context. In the second stage, the model selects the most suitable candidate using a heuristic that favors the point closest to the target. The selected point is appended to the trajectory and fed back into the model until all destinations are reached. This framework combines predictive modeling with iterative inference to enable dynamic, goal-directed route planning.

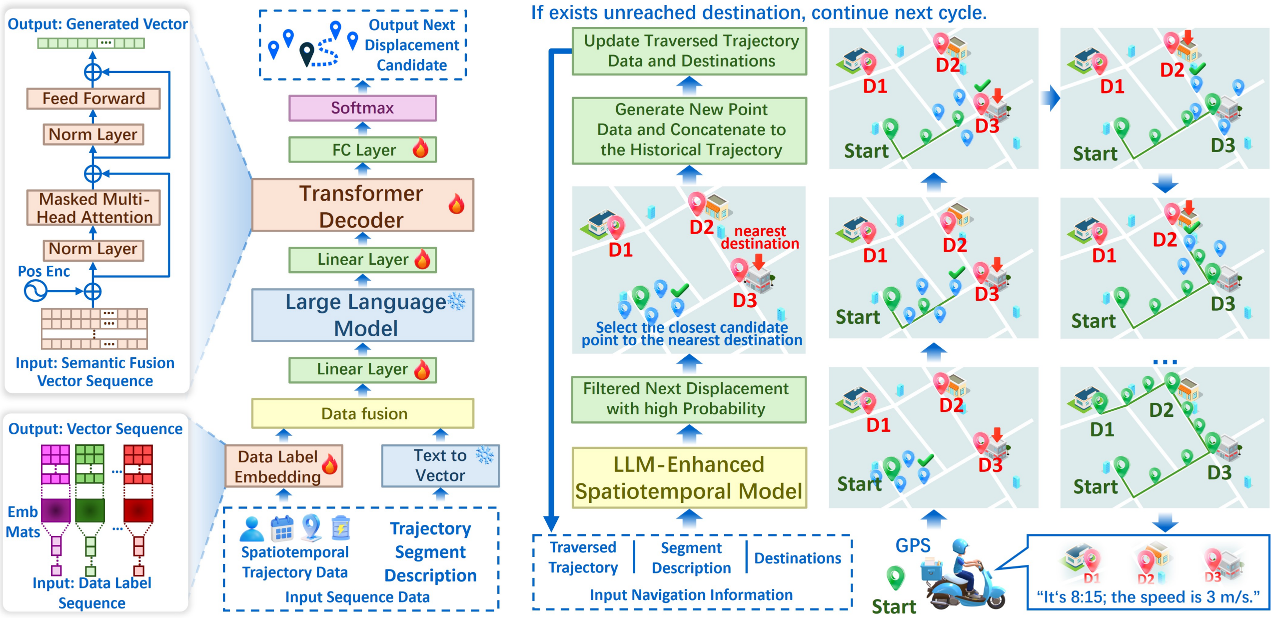
A. Spatiotemporal Trajectory Model

The spatiotemporal trajectory model is the core of the navigation framework, comprising a spatiotemporal-semantic fusion module and a trajectory point prediction module, as illustrated in Figure 1(a). The fusion module integrates spatiotemporal and textual inputs into a unified representation, while the prediction module uses this fused history to forecast the next set of candidate locations.

Semantic Fusion Module. The fusion module, shown in Figure 1(a), maps input sequence data to a fused representation via a pretrained LLM. The input consists of two types. The first type is numerical data, including user, temporal, spatial location, and battery information. These are discrete labels, each embedded into vectors through separate embedding layers, as illustrated in the bottom-left subfigure of Figure 1(a). The second type is textual data that captures semantic descriptions of motion status along the riding trajectory. These textual inputs are vectorized by a text-to-vector module. The numerical embeddings v_{traj} and semantic embeddings v_{text} are then normalized, concatenated, and passed through a linear projection layer to adjust dimensionality before being fed into the LLM:

$$v_f = LLM(concat(v_{traj}, v_{text})) \tag{1}$$

The output v_f is the final fused embedding for the trajectory sequence. During training, both the text-to-vector module and the LLM have frozen parameters and are not updated.



(a) LLM-Enhanced Spatiotemporal Model **(b) Trajectory Inference**

Figure 1: Framework of Talking Trails. The two main processes: displacement prediction via an LLM-enhanced spatiotemporal model and step-by-step trajectory inference, are illustrated in (a) and (b), respectively.

Decoder-only Module. The decoder-only module, as illustrated in Figure 1(a), takes the fused representation from the LLM as input and generates the final output. It is adopted due to its strong capability in modeling sequential data with temporal dependencies.

Given each trajectory node $v_i \in V$, the model utilizes its fused embedding to predict the displacement(direction and distance) to the next position. The decoder-only module mainly relies on a masked multi-head attention mechanism, which applies a causal mask to prevent access to future positions and computes attention scores over the previous vectors v_1,...,v_n to inform the prediction of v_{n+1}. The attention score is calculated as

$$MaskedAttention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}} + M)V, \tag{2}$$

where Q, K, and V represent the query, key, and value matrices extracted from the input trajectory sequence, respectively, and d_k is the dimension of the key vector for scaling. The mask matrix M ensures that the attention mechanism only considers previous positions, thus implementing masked attention. The decoder processes the sequence through N layers of masked multi-head attention layers, iteratively updating the representation by focusing on relevant historical node information and ignoring future positions. The softmax function in the masked multi-head attention converts each raw score of vectors into a probability distribution, ensuring that the sum of all vector is 1.

Loss Function and Objective. The task of displacement prediction based on the current trajectory sequence is defined as a multi-classification problem, where each class represents a potential displacement from the current position. The true label y specifies the actual displacement to the next position. A fully connected layer maps the final fused embeddings to a fixed-size output space, producing one logit z for each class. The final softmax layer converts these logits into probabilities interpretable as prediction confidence.

To train the model, the cross-entropy loss between the predicted probability distribution and the true label y is minimized. Logits z_i represent the model's raw output for the i-th displacement label, corresponding to a specific direction and distance to the next position. The cross-entropy loss is defined as follows:

$$L = -\sum_{i=1}^{n} y_i log(\frac{exp(z_i)}{\sum\limits_{j=1}^{n} exp(z_j)}), \tag{3}$$

which evaluates the model's prediction accuracy of the displacement for an input trajectory sequence.

The loss function minimizes prediction errors by penalizing deviations between predicted probabilities and true labels. Through this optimization, the model improves next-position prediction accuracy and learns route planning patterns from rider experience, ultimately gaining the ability to generate high-precision, efficient delivery routes.

B. Trajectory Inference

The spatiotemporal trajectory model infer a next trajectory point based on the current already generated trajectory points sequence with a heuristic search for optimal distance. The processing of trajectory inference is illustrated as Figure 1(b). When the model receives an trajectory sequence $H_{traj} = \{h_1, h_2, ..., h_n\}$ where h_n represents the last inferred points in the trajectory, and a set of multiple destination points $H_{dest} = \{h_{dest1}, h_{dest2}, ..., h_{destm}\}$, the model generates a set of next point candidates displacement $M = \{m_1, m_2, ..., m_p\}$ based on fused data representations. It then filters the candidate displacements to obtain $M' = \{m_i | p_i > \lambda\}$, where p_i represents the predicted probability of the candidate displacement $m_i \in M$ according to the probability threshold λ. The set of candidate next predicted points, converted to their corresponding location positions, is denoted as H_{next}.

The heuristic search strategy employed here operates as follows: first, among the multiple destination points in H_{dest}, the point $h_{closest}$ with the shortest distance to the current location is selected as the current target. Then, from the set of candidate next positions H_{next}, the model selects the optimal point h_{optim} that minimizes the distance to the target destination.

$$h_{closet} = argmin_{h_d \in H_{dest}} dist(h_n, h_d) \tag{4}$$

$$h_{optim} = argmin_{h_m \in H_{next}} dist(h_{closest}, h_{dest}) \tag{5}$$

The selected trajectory point h_{optim} will be dynamically added to the trajectory sequence for a planning route.

$$H'_{traj} = \{h_1, h_2, ..., h_n, h_{optim}\} \tag{6}$$

A route planning task is completed when all destination points have been removed from H_{dest}, and H'_{traj} is the generated delivery route. During the point-by-point generation process, we assume that all non-location feature dimensions take default values based on typical training data. For special scenarios, these default values can be adjusted accordingly to better match the specific context.

Experiments

In this section, we provide a detailed description of the training data, evaluation metrics, experimental setup, and results for the spatiotemporal model in Talking Trails.

Dataset

The training data for Talking Trails were sourced from the battery swapping IoT system deployed by Hangzhou Yugu Technology Co., Ltd. The dataset contains approximately 1 million real-world e-bike delivery trajectories, primarily collected between July and September 2024 in central urban areas of Hangzhou, China. Trajectories were recorded via the e-bike battery's central control system and are predominantly from experienced food delivery riders, whose average daily trajectory length exceeds 120 km, capturing high-activity periods and locations. Each trajectory sequence averages 25 location points, corresponding to a travel distance of 4-5 km. The raw data include user identifiers, temporal attributes, spatial coordinates, and battery status (voltage and capacity). To ensure data quality, trajectories with significant GPS loss, prolonged stops, repeated drifts, or extended inactivity were removed. During preprocessing, data were segmented and enriched with annotations: user attributes (gender, age), temporal tags (month, weekday, time-of-day), and spatial encoding using the H3 geospatial indexing system (resolution 10). Displacements between consecutive points were represented as relative coordinates derived from H3 codes. Additionally, semantic text descriptions of riding status (e.g., "It is 8:16. The next stop is 89 meters away, estimated to take 30 seconds at 3.0 m/s") were dynamically generated from displacement and timestamp data. These textual inputs were vectorized using the bge-small-zh v1.5 semantic embedding model to enhance training efficiency. The final training dataset integrates structured spatiotemporal features with vectorized semantic representations. Model performance was evaluated on a test set of 10,000 trajectory sequences randomly sampled from the full dataset.

Evaluation Metrics

We evaluate the proposed semantic-enhanced delivery route planning method using three metrics. The definitions of these metrics are provided as below.

- Point Prediction Accuracy. This metric assesses the model's ability to predict the displacement in a trajectory sequence, reflecting its capacity to capture movement patterns. We use Top-1 Accuracy (Acc@1) and Top-5 Accuracy (Acc@5): a prediction is correct if the true displacement is ranked first or within the top five predicted candidates.
- Arrival Rate (Arr.). It evaluates whether a generated trajectory successfully reaches the destination. The trajectory is deemed failed if its length exceeds $\sqrt{2}$ times that of the original trajectory (Li et al. 2025), or if it fails to avoid both impassable locations and consecutive crossings of direct shortcut paths.
- Optimal Rate (Opt.). It evaluates the performance of Talking Trails in generating distance-optimized routes. Given the same origin and destination, a generated route is regarded as optimal if its total distance is shorter than that of the actual rider's trajectory. This metric reflects the model's capability to shorten and optimize routes.

Experiment Setup

The spatiotemporal trajectory model is implemented in PyTorch and optimized using the Adam optimizer with a learning rate of 0.001. The multi-head masked attention mechanism employs 8 attention heads. For semantic representation, the PLM bge-small-zh-v1.5 is used to generate 512-dimensional vector embeddings for the textual description of each trajectory segment. The Llama-3.2-1B LLM is then employed to fuse the trajectory data embeddings with the semantic embeddings, chosen as the optimal balance between performance and inference speed to ensure per-trajectory generation latency remains at the sub-second level for practical user experience. To investigate the influence of key design choices, we conduct the following three experiments:

- **Ablation of the Semantic Fusion Module.** To assess the role of the semantic fusion module, we compare the model's performance in route planning and displacement prediction with and without this component. Specifically, the model variant without the semantic fusion module neither takes semantic trajectory descriptions as input nor employs an LLM to integrate semantic embeddings with trajectory embeddings.

- **Comparison of Semantic Fusion Methods.** The choice of data fusion method can substantially affect how semantic information is incorporated into the model, thereby impacting overall performance. To examine this effect, we compare two common fusion strategies, vector addition and vector concatenation, and evaluate their effectiveness in the context of semantic-enhanced trajectory modeling.
- **Impact of Model Parameter Sizes.** Since the size of trainable model parameters typically affects both computational efficiency and overall performance, we evaluate the model under different parameter scales (i.e., 30M, 50M, and 100M) for the trainable modules, as illustrated in Figure 1(a).

Results and Analysis

The experimental results are summarized in Table 1. The ablation study (E3 vs. E1) shows that the semantic data fusion module significantly improves model performance, with its removal leading to drops of 13.8% in Top-1 accuracy, 8.8% in Top-5 accuracy, 21% in arrival rate, and 15.1% in optimal rate. This highlights its critical role in capturing contextual and spatial patterns. Among fusion strategies, vector concatenation (E3) outperforms vector addition (E2), achieving gains of 5.3%, 2.5%, 4.6%, and 0.5% across the four metrics, respectively, attributed to its better preservation of feature distinctiveness. In the model size comparison, the 100M-parameter model (E5) delivers the best results, outperforming the 30M model (E3) by up to 3.1% and the 50M model (E4) by up to 1.8% in Top-5 accuracy. We did not further scale the model beyond 100M, as larger models showed only marginal performance gains. Therefore, we selected the 100M model as the optimal configuration for our current dataset, and it was ultimately deployed in the Talking Trails application.

To evaluate the route optimization performance of Talking Trails under real-world scenarios, we compare the generated routes with actual delivery trajectories across four different scenarios, as shown in Figure 2. In Group A, we can observe that Talking Trails can effectively identify narrow bridge roads accessible to electric bicycles, and the generated optimal route is shortened by 5.75%. This indicates that Talking Trails has the capability to identify shortcuts and alternative paths via learning the rider's route planning experience. In Group B, the model generated a route across residential areas by navigating through alleys and pathways within communities to reach the destination. This demonstrates the model's capability to effectively navigate complex road environments in residential areas, thereby improving delivery efficiency. In Group C, the model generated a route to minimize detours in complex road conditions, which reduced the total distance by 27.27%. This illustrates that the model can flexibly apply the riders' route planning experience learned from real delivery trajectories. In Group D, the model generated a route to the real trajectory does not involve detours, which reduced the total distance by 6.1%. Statistical analysis on the validation set shows that Talking Trails can effectively shorten delivery distances, with an average trajectory length of 4.8 km and an average saving of 0.84 km, which represents a 17% reduction in riding distance, thereby enhancing delivery efficiency and bringing significant benefits to riders.

Index	Size	Fusion	Acc@1	Acc@5	Arr.	Opt.
E1	30M	none	43.8%	83.7%	69.5%	65.6%
E2	30M	add	52.3%	90.0%	85.9%	80.2%
E3	30M	concat	57.6%	92.5%	90.5%	80.7%
E4	50M	concat	60.1%	93.8%	91.4%	81.3%
E5	100M	concat	**60.2%**	**95.6%**	**91.5%**	**82.1%**

Table 1: Performance Comparison of Different Models

Talking Trails Application

This section introduces the application of Talking Trails, including its integration into the semantic-to-semantic service pipeline, as well as its implementation and performance in battery-swap interfaces.

A. Semantic-to-Semantic User Service

To better deploy this service, we adopted an open-source LLM to enable semantic-to-semantic interaction with users. By using prompt engineering and connecting the LLM to databases related to geographic information and traffic conditions, we significantly enhanced its ability to extract input information and enriched the descriptive details of the output results. A schematic diagram of the overall semantic-to-semantic interaction process is shown in Figure 3(a). Initially, the rider inputs multiple delivery destinations by articulating their intentions via voice input to the terminal device. The device then processes the spoken input, converting it into semantic text, which is uploaded to the cloud server for further analysis. The semantic text is subsequently parsed by the open-source LLM, which accurately interprets the rider's intent. Once classified as a navigation or delivery task, the LLM extracts key geographical locations and the rider's other related features. These extracted elements, such as the starting and ending points, time, battery level, and riding speed, are converted into structured spatiotemporal data labels and corresponding textual descriptions, making them compatible for processing by the Talking Trails system. For any missing information, the system supplements it based on the user's registered data or default historical reference information. After the trajectory is generated by the Talking Trails system, the system associates the spatiotemporal points of the trajectory with a self-constructed geographic and traffic information database. It then retrieves textual descriptions relevant to the generated trajectory. These retrieved texts are subsequently refined and reorganized by an open-source LLM to provide users with important road condition information and cautions associated with the planned route. The resulting textual content is transmitted back to the user's terminal and

(a) Group A **(b) Group B** **(c) Group C** **(d) Group D**

Figure 2: Comparison of Talking Trails generated trajectories versus original trajectories with identical start and end points. (a)–(d) show four example pairs from the validation set. In each pair, the top panel depicts the original trajectory and the bottom panel the generated trajectory.

converted into voice feedback, while the trajectory itself is presented to the user through a visualized map interface.

B. Interface for Battery-Swap Scenarios

Battery-swap for e-bikes is a convenient method of energy replenishment, allowing users to quickly exchange depleted batteries for fully charged ones at swapping stations when their battery power is low, enabling an immediate ”swap-and-go” process without the need for lengthy charging waits. This model is particularly suitable for high-frequency battery usage scenarios such as food delivery, courier services, and daily commuting. Currently, this approach has been widely adopted in the food delivery industry. To better serve food delivery riders, we have integrated Talking Trails into the mobile app “Zheli Battery Swap”[1], providing real-time path planning and semantic interpretation services for delivery riders. To ensure that navigation routes in Talking Trails lead riders to their actual destinations in real-world scenarios rather than following theoretical road networks alone, we have integrated real-time battery level data and battery swapping station information into the system. After generating the delivery trajectory, the system estimates energy consumption based on the initial battery volume and converts the remaining capacity into an estimated travel-able distance. A battery threshold of 30% is set to trigger swapping decisions. If the battery volume is predicted to fall below this threshold along the planned route, the system proactively includes nearby battery swapping stations as intermediate targets, ensuring that riders can complete their delivery tasks without interruption due to insufficient battery energy. This functionality helps alleviate riders’ range anxiety and ensures smooth and uninterrupted delivery performance.

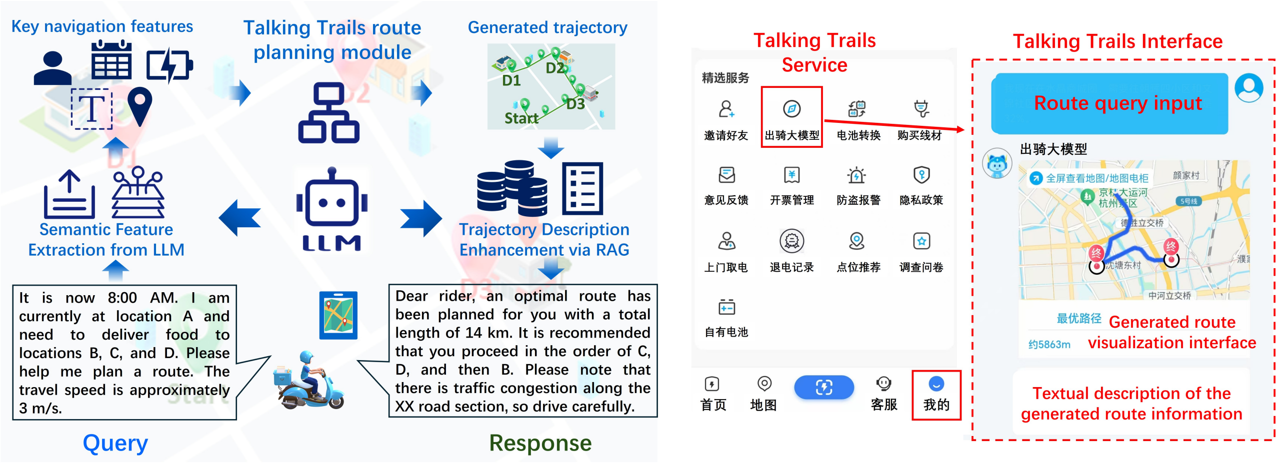
Figure 3(b) shows the battery swapping interface and the usage steps of Talking Trails. After logging into the app’s main interface, users navigate to the ”My” tab and select Talking Trails. They then input a route-related query (e.g., specifying origin and destination), and the system responds with a visual representation of the generated route along with a textual description of the route information. The returned description includes details on destination planning, total distance, estimated travel time, road conditions along the route, and safety advisories such as hazardous intersections or areas requiring heightened awareness.

Talking Trails Deployment

Our system employs a modular deployment architecture, which divides the overall service into five functional components: a foundational LLM API, a trajectory semantic description API, a geolocation decoding API, the path generation module Talking Trails, and a service orchestration and integration API. The foundational language model module is built on Qwen-14B, serving as the unified semantic processing backbone that supports natural language tasks such as path interpretation and user interaction. The trajectory semantic description module follows the Retrieval-Augmented Generation (RAG) paradigm, leveraging structured labels from trajectory points to perform label-level matching with a pre-constructed geographic and traffic knowledge base. These matched elements are then combined with the language model to generate context-aware natural language descriptions, thereby enhancing the interpretability and semantic richness of the generated paths. The geolocation decoding module converts user-provided geographic text into standardized geocodes (the label corresponding to h3 geo-code), improving the accuracy of location recognition. The Talking Trails module dynamically generates complete delivery paths

---

[1] A mobile application for e-bike battery swapping, developed by Hangzhou Yugu Technology Co., Ltd. App download link: https://yuguadc.yugu.net.cn/h5_download_page/index_tarck.html?page=downloadAppchannel=mpstatus=0attributes=type=1



It is now 8:00 AM. I am currently at location A and need to deliver food to locations B, C, and D. Please help me plan a route. The travel speed is approximately 3 m/s.

Dear rider, an optimal route has been planned for you with a total length of 14 km. It is recommended that you proceed in the order of C, D, and then B. Please note that there is traffic congestion along the XX road section, so drive carefully.

(a) Semantic-to-Semantic Service

(b) Interface for Battery-Swap Scenarios

Figure 3: Application of Talking Trails. (a) illustrates the complete pipeline, from user's semantic navigation queries and feature extraction by LLM, to Talking Trails route planning, RAG-based route description generation, and finally to the application's response. (b) shows the Talking Trails application on the battery swap interface.

by integrating trajectory prediction models with rule-based path planning algorithms, conditioned on the rider's departure point and task-specific preferences. To improve the robustness of path generation in navigation, we adopt a map-based road network navigation API as a fallback mechanism when the primary path generation fails. The integration API orchestrates the service flow as illustrated in Figure 3(a), exposing a unified interface for consumption by the mobile application. The entire system is deployed on a cluster equipped with NVIDIA A100 GPUs.

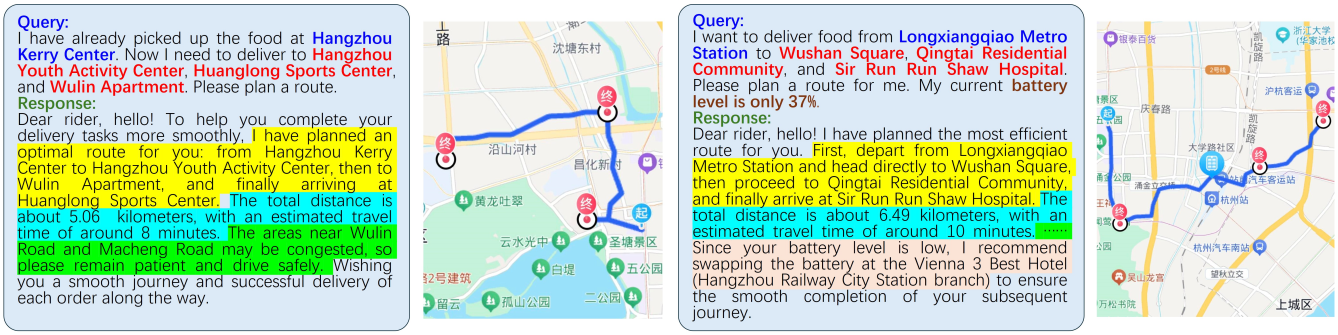
Talking Trails is currently deployed in Hangzhou, serving tens of thousands of delivery riders across the city's urban area and providing real-time route planning and semantic interaction services. Based on operational data, each delivery rider travels an average of approximately 120 kilometers per day. According to the path optimization rate of 17% for the overall rider group, as calculated in our experimental section, we can estimate that, on average, a rider can save about 20.4 kilometers in daily travel distance after using Talking Trails. Considering the average riding speed of 35 kilometers per hour, each rider can save around 35 minutes per day under unchanged delivery tasks, significantly improving delivery efficiency. These time and distance savings translate into reduced operational fatigue for riders and increased delivery capacity for service platforms.

Furthermore, the reduction in travel distance helps lower riders' carbon emissions. Based on the conversion standard of 1 kWh of electricity enabling a 40-kilometer ride and corresponding to 0.39 kilograms of carbon emissions (Ding et al. 2025), using Talking Trails reduces approximately 199 grams of carbon emissions per rider each day. Assuming 250 working days per year, this translates to about 49.7 kilograms of carbon emissions saved per rider annually. Currently, Talking Trails is available to approximately 80,000 riders in Hangzhou, with an estimated annual reduction of around 3,978 tons of carbon emissions for the entire city. This carbon emission reduction contributes to urban sustainability by improving air quality, mitigating climate change, and promoting public health, delivering tangible environmental and societal benefits at scale. Overall, these results demonstrate that Talking Trails holds strong potential to make a substantial contribution to both delivery efficiency and carbon emissions reduction at the city scale.

Case Study

To demonstrate the improvements in user experience after the deployment of Talking Trails, we illustrate its convenience and efficiency for riders through four typical application scenarios (as shown in Figure 4). When users ask Talking Trails a question, the LLM first extracts key information from the query, with the keywords highlighted in bold in the inquiry texts of Figure 4. The main elements include the user's current location, destination(s), time information, and battery level. Among these, the destination is the core piece of information that must be provided; the starting point and time information can be automatically obtained from mobile hardware; and the battery level is used to determine whether a route with battery swap stations needs to be planned. If the user does not mention battery level, the system defaults to normal route planning. If battery information is included, the system estimates mileage to determine whether the battery meets the safety threshold and, when necessary, recommends battery swap stations.

Based on the user's query, Talking Trails dynamically adjusts its responses, which mainly include the following components: first, the basic route planning, suggesting the sequence of multiple destinations (indicated with a yellow background in Figure 4 responses); second, the total trip length and estimated travel time (cyan background); third, traffic conditions along the route, derived from the road database, to help riders anticipate road situations and reduce the likelihood of accidents (green background); and fourth, for queries

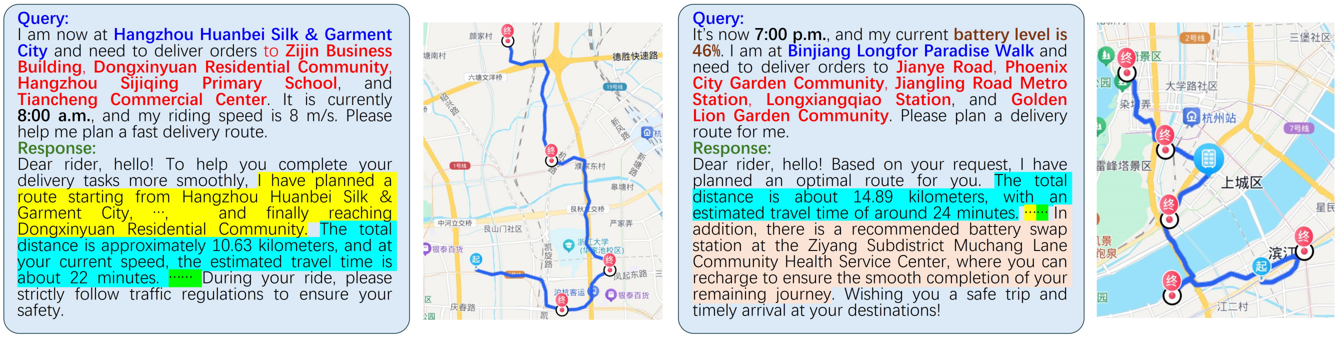


Query:
I have already picked up the food at **Hangzhou Kerry Center**. Now I need to deliver to **Hangzhou Youth Activity Center**, **Huanglong Sports Center**, and **Wulin Apartment**. Please plan a route.
Response:
Dear rider, hello! To help you complete your delivery tasks more smoothly, I have planned an optimal route for you: from Hangzhou Kerry Center to Hangzhou Youth Activity Center, then to Wulin Apartment, and finally arriving at Huanglong Sports Center. The total distance is about 5.06 kilometers, with an estimated travel time of around 8 minutes. The areas near Wulin Road and Macheng Road may be congested, so please remain patient and drive safely. Wishing you a smooth journey and successful delivery of each order along the way.

Query:
I want to deliver food from **Longxiangqiao Metro Station** to **Wushan Square**, **Qingtai Residential Community**, and **Sir Run Run Shaw Hospital**. Please plan a route for me. My current **battery level is only 37%**.
Response:
Dear rider, hello! I have planned the most efficient route for you. First, depart from Longxiangqiao Metro Station and head directly to Wushan Square, then proceed to Qingtai Residential Community, and finally arrive at Sir Run Run Shaw Hospital. The total distance is about 6.49 kilometers, with an estimated travel time of around 10 minutes. ······ Since your battery level is low, I recommend swapping the battery at the Vienna 3 Best Hotel (Hangzhou Railway City Station branch) to ensure the smooth completion of your subsequent journey.

(a) Case 1: Short-distance navigation without battery swap

(b) Case 2: Short-distance navigation with battery swap



Query:
I am now at **Hangzhou Huanbei Silk & Garment City** and need to deliver orders to **Zijin Business Building**, **Dongxinyuan Residential Community**, **Hangzhou Sijiqing Primary School**, and **Tiancheng Commercial Center**. It is currently **8:00 a.m.**, and my riding speed is 8 m/s. Please help me plan a fast delivery route.
Response:
Dear rider, hello! To help you complete your delivery tasks more smoothly, I have planned a route starting from Hangzhou Huanbei Silk & Garment City, ···, and finally reaching Dongxinyuan Residential Community. The total distance is approximately 10.63 kilometers, and at your current speed, the estimated travel time is about 22 minutes. ······ During your ride, please strictly follow traffic regulations to ensure your safety.

Query:
It's now **7:00 p.m.**, and my current **battery level is 46%**. I am at **Binjiang Longfor Paradise Walk** and need to deliver orders to **Jianye Road**, **Phoenix City Garden Community**, **Jiangling Road Metro Station**, **Longxiangqiao Station**, and **Golden Lion Garden Community**. Please plan a delivery route for me.
Response:
Dear rider, hello! Based on your request, I have planned an optimal route for you. The total distance is about 14.89 kilometers, with an estimated travel time of around 24 minutes. ······ In addition, there is a recommended battery swap station at the Ziyang Subdistrict Muchang Lane Community Health Service Center, where you can recharge to ensure the smooth completion of your remaining journey. Wishing you a safe trip and timely arrival at your destinations!

(c) Case 3: Long-distance navigation without battery swap

(d) Case 4: Long-distance navigation with battery swap

Figure 4: Four representative delivery cases illustrating short-distance and long-distance routes with/without battery swapping. In the queries, keywords are bolded, while blue, red, and brown text denote origin, destination, and battery status. In the Talking Trails responses, route, distance/time, traffic, and swap points are highlighted with yellow, cyan, green, and pink backgrounds, respectively. Excess details are omitted with ellipses.

involving battery status, information on recommended battery swap points (pink background), ensuring riders can successfully complete their delivery tasks.

Specifically, Figure 4(a) shows the most common scenario of a short-distance delivery without battery swapping, carried out in the busy downtown area of Hangzhou, highlighting route optimization in complex alleys and pedestrian streets. Figure 4(b) presents a similar delivery scenario in the same region but with the added requirement of battery swapping, demonstrating how the system ensures sufficient power by incorporating the nearest swap station. Figure 4(c) depicts a longer-distance delivery that takes into account both the current time and speed requirements, showcasing how the system adjusts riding recommendations based on these factors. Finally, Figure 4(d) demonstrates a crowdsourced rider facing multiple long-distance destinations, emphasizing the importance of battery management and showing how the system intelligently selects the optimal swap station. Together, these four cases not only demonstrate the functional features of Talking Trails but also highlight its capability to handle various types of delivery tasks.

Together, these cases demonstrate Talking Trails' ability to support diverse delivery tasks through LLM-enhanced semantic understanding and spatiotemporal modeling. By enabling end-to-end semantic interaction, dynamic battery swapping support, and context-aware routing, the system delivers a rider-centric experience, improves route viability, reduces delivery anxiety, and bridges the experience gap across urban environments. It demonstrates the potential for intelligent, efficient, and sustainable instant delivery.

Conclusion

This paper presents Talking Trails, a semantic-enhanced spatiotemporal trajectory modeling framework for e-bike delivery route planning. Trained on massive real-world trajectory data, the model leverages LLM to integrate spatiotemporal, battery, and textual semantic information, achieving a top-5 rider displacement prediction accuracy of 95% and a route optimization rate of 82.1%. By incorporating LLM-based task augmentation and introducing battery-swap support, Talking Trails has achieved practical application, providing efficient navigation services to delivery riders. Currently deployed for over 80,000 riders in Hangzhou, Talking Trails is expected to reduce average delivery mileage by 17% and cut approximately 4,000 tons of carbon dioxide emissions annually, making significant contributions to improving urban operational efficiency and advancing sustainable urban development.

Acknowledgments

This work was supported by the National Key R&D Program of China (Grant No. 2023YFB4502400) and the National Natural Science Foundation of China (Grant No. 62176236).

References

Alhussen, A.; and Ansari, A. S. 2024. Real-Time Prediction of Urban Traffic Problems Based on Artificial Intelligence-Enhanced Mobile Ad Hoc Networks (MANETS). *Computers, Materials & Continua*, 79(2).

Amin, A.; Amin, M. S.; and Cho, C. 2023. An Application to Predict Range of Electric Two-Wheeler Using Machine Learning Techniques. *Applied Sciences*, 13(10).

Boczka, B.; Pluzsik, M.; Mocsai, T.; Botzheim, J.; and Reményi, I. 2025. Application of a Deep Learning-Based Heuristic for Weighted A-Star Algorithm. In *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, 145–149.

Chen, W.; Liang, Y.; Zhu, Y.; Chang, Y.; Luo, K.; Wen, H.; Li, L.; Yu, Y.; Wen, Q.; Chen, C.; et al. 2024. Deep learning for trajectory data management and mining: A survey and beyond. *arXiv preprint arXiv:2403.14151*.

Ding, B.; Yu, J. X.; and Qin, L. 2008. Finding time-dependent shortest paths over large graphs. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, 205–216.

Ding, D.; Li, Z.; Zhang, J.; Liu, X.; Zhang, J.; Li, Y.; Cai, P.; Liu, J.; and Long, G. 2025. eBaaS: AIoT-Enabled eBike Battery-Swap as a Service for Last-Mile Delivery. In *Proceedings of the ACM on Web Conference 2025*, 5045–5053.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36: 19622–19635.

Hsu, C.-C.; Kang, L.-W.; Chen, S.-Y.; Wang, I.-S.; Hong, C.-H.; and Chang, C.-Y. 2023. Deep learning-based vehicle trajectory prediction based on generative adversarial network for autonomous driving applications. *Multimedia Tools and Applications*, 82(7): 10763–10780.

Kanoulas, E.; Du, Y.; Xia, T.; and Zhang, D. 2006. Finding fastest paths on a road network with speed patterns. In *22nd International Conference on Data Engineering (ICDE'06)*, 10–10. IEEE.

Koenig, S.; and Likhachev, M. 2002. D* lite. In *Eighteenth national conference on Artificial intelligence*, 476–483.

Lee, B. C.; Han, S. R.; and Song, B. D. 2023. The Use of Electric Cargo Bike and Cycle for Sustainable Last Mile Delivery: Optimization and Evaluation. *SSRN*.

Li, L.; Yao, J.; Wenliang, L.; He, T.; Xiao, T.; Yan, J.; Wipf, D.; and Zhang, Z. 2021. GRIN: Generative Relation and Intention Network for Multi-agent Trajectory Prediction. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 27107–27118. Curran Associates, Inc.

Li, Z.; Jiao, Y.; Shi, Y.; Ding, D.; Wang, J.; Zhang, J.; and Xu, H. 2025. RideSmart: Pre-trained Large Models for Delivery Route Planning. In *Companion Proceedings of the ACM on Web Conference 2025*, 2867–2870.

Li, Z.; Xia, L.; Tang, J.; Xu, Y.; Shi, L.; Xia, L.; Yin, D.; and Huang, C. 2024. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5351–5362.

Liu, Y.; Qin, G.; Huang, X.; Wang, J.; and Long, M. 2024. AutoTimes: autoregressive time series forecasters via large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, 122154–122184.

Lu, E. H.-C.; Chen, H.-S.; and Tseng, V. S. 2016. An efficient framework for multirequest route planning in urban environments. *IEEE Transactions on Intelligent Transportation Systems*, 18(4): 869–879.

Ma, H.; Sun, Y.; Li, J.; Tomizuka, M.; and Choi, C. 2021. Continual multi-agent interaction behavior prediction with conditional generative memory. *IEEE Robotics and Automation Letters*, 6(4): 8410–8417.

Nourmohammadi, Z.; Rey, D.; and Saberi, M. 2025. E-Cargo Bike Route Optimization with Rider Fatigue Considerations: A Chance-Constrained Programming Approach. *Available at SSRN 5142171*.

Panagoulias, D. P.; Virvou, M.; and Tsihrintzis, G. A. 2024. Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis. *Electronics*, 13(2): 320.

Sadid, H.; and Antoniou, C. 2024. Dynamic spatio-temporal graph neural network for surrounding-aware trajectory prediction of autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*.

Salzmann, T.; Ivanovic, B.; Chakravarty, P.; and Pavone, M. 2020. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 683–700. Cham: Springer International Publishing.

Stentz, A. 1994. Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE international conference on robotics and automation*, 3310–3317. IEEE.

Sun, C.; Li, H.; Li, Y.; and Hong, S. 2024. TEST: Text Prototype Aligned Embedding to Activate LLM's Ability for Time Series. In *The Twelfth International Conference on Learning Representations*.

Wu, T.; and Ling, Q. 2024. STELLM: Spatio-temporal enhanced pre-trained large language model for wind speed forecasting. *Applied Energy*, 375: 124034.

Xue, H.; and Salim, F. D. 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11): 6851–6864.

Ye, J.; Gou, Y.; Liang, H.; Yuan, F.; and Yang, C. 2025. A Design Method for Shared Two-Wheeled Electric Scooters (STWESs), Integrating Context Theory and Kansei Engineering. *Sustainability*, 17(8).

Zhang, L.; Zhu, X.; and Ma, J. 2023. IoT Route Planning Based on Spatiotemporal Interactive Attention Neural Network. *IEEE Internet of Things Journal*, 11(5): 7697–7709.