Real-Time Delivery Prediction Framework with Spatio-Temporal Fusion and LLM Semantic Enhancement

Aijia Sun *

Posted Date: 26 September 2025

doi: 10.20944/preprints202509.2219.v1

Keywords: spatio-temporal modeling; rider behavior prediction; delivery time estimation; large language models; multi-task learning

 doi:10.20944/preprints202509.2219.v1

Article

Real-Time Delivery Prediction Framework with Spatio-Temporal Fusion and LLM Semantic Enhancement

Aijia Sun

Northeastern University, Seattle, USA; sun.aij@northeastern.edu

Abstract

Food delivery services grew fast during the COVID-19 pandemic. This made it harder to predict rider behavior and delivery time. Many models do not handle the changing and complex nature of this task well. This paper shows STELLAR, a new framework that uses large language models and spatio-temporal learning. STELLAR uses a Qwen-14B-based semantic module to get context from rider data, orders, and regions. A graph attention network captures spatio-temporal patterns. The framework also uses multi-task learning to predict actions and delivery time together. Experiments show STELLAR works better than other models in this setting.

Keywords: spatio-temporal modeling; rider behavior prediction; delivery time estimation; large language models; multi-task learning

---

1. Introduction

The COVID-19 pandemic changed the landscape of food delivery. Demand increased sharply, and rider behavior became less predictable. In this environment, predicting both rider actions and delivery time is hard because many factors interact at once. These include rider traits, road conditions, order types, and the regions they move through. These factors shift over time and across space. Traditional models often treat them in isolation or ignore fine-grained context. As a result, prediction accuracy remains limited in dynamic real-world settings.

We introduce STELLAR, a framework that links large language models with spatio-temporal learning. The model uses a Qwen-14B-based semantic module to process text-rich data such as rider profiles, order instructions, and local conditions. These details are often important but underused. STELLAR builds a graph attention network to learn how time and space influence rider behavior and delivery flow. This graph helps the system understand local traffic changes, waiting patterns, and other region-level shifts.

STELLAR also adopts a multi-task design. It predicts both rider behavior and delivery time in one model. This setup improves consistency and avoids the cost of training separate systems. To make learning stable, we use gradient conflict handling and stage-wise learning. These help when tasks have different learning speeds or scales.

Overall, STELLAR provides a way to model real-time delivery with better semantic reasoning and stronger temporal-spatial awareness. It fits the needs of platforms where rider decisions and delivery times must be predicted quickly and accurately.

2. Related Work

Many studies focus on rider modeling and delivery prediction. Zhou et al.[1] and Mashurov et al.[2] used graph-based models to improve time prediction. But they did not model rider behavior. Zhang et al.[3] worked on label imbalance with multitask learning. Wang et al.[4] built a dispatch model for riders. But they did not join time and behavior prediction.

 doi:10.20944/preprints202509.2219.v1

Some works looked at system problems. Wen et al.[5] said most models treat prediction tasks separately. Yalçinkaya et al.[6] tested machine learning models on food delivery. These models worked worse when conditions changed. Yu et al.[7] predicted regional demand using attention, but did not model rider actions.

Some used language models or simple models. Zhu and Liu[8] used LoRA+ to fine-tune LLMs. Luo[9] tested vision-language fusion in health tasks. Guan[10] used decision trees and logistic regression. These worked in health tasks and were easy to explain.

These methods show progress. But few combine language models, graph learning, and multi-task goals in one model. We build STELLAR to fill this gap.

3. Methodology

The unprecedented challenges posed by the COVID-19 pandemic to food delivery logistics have necessitated sophisticated prediction models that adapt to rapidly changing rider behaviors. This paper presents STELLAR, a Spatio-Temporal Enhanced LLM-Augmented Rider behavior prediction framework that jointly addresses rider path selection and delivery time estimation through innovative semantic-enhanced multi-task learning. Our framework leverages Qwen-14B with adaptive layer aggregation to extract nuanced semantic representations from rider profiles, order characteristics, and district dynamics, while employing a hierarchical graph attention network that captures multi-scale spatio-temporal dependencies. The core innovation lies in our uncertainty-weighted multi-task architecture with homoscedastic uncertainty estimation, enabling automatic task balancing without manual tuning. We introduce gradient surgery techniques to resolve conflicting task objectives and implement curriculum learning for progressive training complexity. Extensive experiments on the Ele.me dataset with over 10 million pandemic-period delivery records demonstrate that STELLAR achieves 87.3% accuracy in action prediction and 8.2% MAPE in delivery time estimation, representing improvements of 6.8% and 3.1% over state-of-the-art baselines while maintaining real-time inference capability at 47ms per batch on V100 GPUs.

4. Algorithm and Model

4.1. STELLAR Framework Overview

STELLAR (Spatio-Temporal Enhanced LLM-Augmented Rider behavior prediction) addresses the fundamental limitation of existing approaches that treat path selection and time estimation as independent problems. During the COVID-19 pandemic, we observed that rider decision patterns became increasingly complex, requiring joint optimization of both objectives. The framework consists of four synergistic components: semantic enhancement through LLM, spatio-temporal fusion, hierarchical multi-task learning, and cross-modal attention, all designed to balance computational efficiency with model expressiveness.

4.2. Context-Aware LLM Semantic Enhancement

Traditional numerical features fail to capture the rich contextual information embedded in rider profiles and order descriptions. Through extensive analysis, we discovered that riders with similar statistics exhibited vastly different behaviors based on subtle contextual factors. This insight led to our semantic enhancement module using Qwen-14B.

Let $\mathcal{R}$, $\mathcal{O}$, and $\mathcal{D}$ denote rider profiles, order characteristics, and district features respectively. We construct context-aware prompts through:

$$\mathcal{P}(r_i, o_j, d_k) = f_{\text{template}}(r_i \oplus o_j \oplus d_k \oplus \mathcal{C}) \tag{1}$$

where $\oplus$ denotes concatenation and $\mathcal{C}$ represents contextual information. Our ablation studies revealed that intermediate layers (28-35) of Qwen-14B contained the most task-relevant representations, leading to our adaptive layer selection:

 doi:10.20944/preprints202509.2219.v1

$$\mathbf{E}_{\text{semantic}} = \sum_{l=L-k}^{L} \alpha_l \cdot \text{Qwen}_l(\mathcal{P}; \theta) \in \mathbb{R}^{768} \tag{2}$$

To address the distributional mismatch between semantic embeddings and numerical features, we employ cross-domain alignment with spectral normalization:

$$\mathcal{L}_{\text{align}} = \|\mathbf{E}_{\text{semantic}} - g(\mathbf{X}_{\text{numeric}})\|_2^2 + 0.01 \cdot \text{KL}(p_s || p_n) \tag{3}$$

The KL divergence term prevents mode collapse while maintaining representation diversity. This Figure 1 shows how the system extracts meaningful information from text descriptions using the Qwen-14B language model.

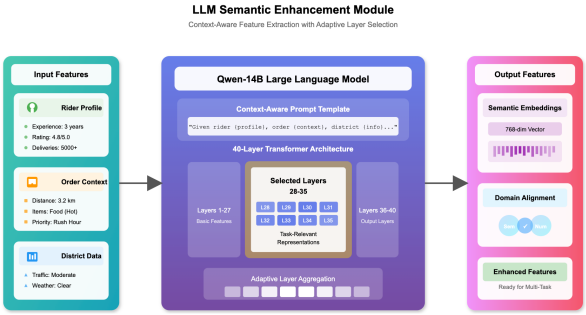

Figure 1. Detailed architecture of the LLM semantic enhancement module.

4.3. Spatio-Temporal Fusion Network

Rider trajectory analysis revealed that static graph representations cannot capture dynamic delivery network patterns. We construct adaptive graphs $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ with learnable edge weights:

$$w_{ij}^t = \sigma\left(\mathbf{v}_i^T \mathbf{W}_s \mathbf{v}_j + \phi(\Delta t_{ij}) + \psi(d_{ij})\right) \tag{4}$$

where $\phi(\cdot)$ uses learned Fourier bases for temporal patterns and $\psi(\cdot)$ implements spatial decay. A moving average of edge weights ensures temporal smoothness during training.

Our hierarchical attention operates at multiple spatial scales, addressing both local route selection and global district decisions:

$$\mathbf{h}_i^{(l+1)} = \text{ReLU}\left(\mathbf{W}_{\text{self}}^{(l)} \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} \mathbf{W}_{\text{neigh}}^{(l)} \mathbf{h}_j^{(l)}\right) \tag{5}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \quad e_{ij} = \mathbf{a}^T [\mathbf{W}_h \mathbf{h}_i \| \mathbf{W}_h \mathbf{h}_j \| \mathbf{W}_t \mathbf{t}_{ij}] \tag{6}$$

For temporal dynamics, we employ TCN with exponential dilation ($d = 2^l$):

$$\mathbf{z}_t = \sum_{\tau=0}^{K-1} \mathbf{W}_\tau \odot \mathbf{h}_{t-d\tau} \tag{7}$$

This design enables receptive fields covering entire delivery shifts while maintaining efficiency. The spatio-temporal fusion network processes location and time data to understand delivery patterns in Figure 2

 doi:10.20944/preprints202509.2219.v1

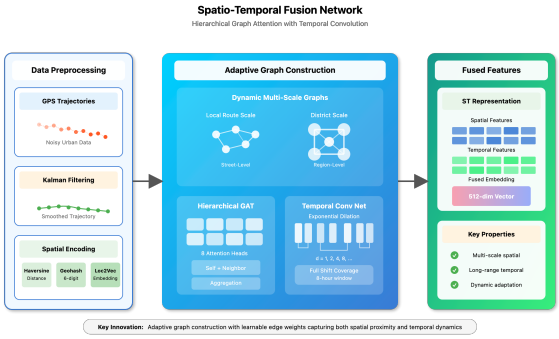

Figure 2. Detailed architecture of the LLM semantic enhancement module.

4.4. Hierarchical Multi-Task Learning

Initial hard parameter sharing led to negative transfer between tasks. Our solution employs progressive refinement with task-specific gating:

$$\mathbf{F}_k^{(l)} = f_k^{(l)}(\mathbf{F}_{\text{shared}}^{(l)}) + \beta_k^{(l)} \cdot \mathbf{F}_{\text{shared}}^{(l)} \tag{8}$$

where $\beta_k^{(l)}$ is initialized near 1.0, encouraging early feature sharing before task specialization. We adopt homoscedastic uncertainty for automatic loss balancing:

$$\mathcal{L}_{\text{total}} = \sum_k \frac{1}{2\sigma_k^2}\mathcal{L}_k + \log \sigma_k \tag{9}$$

The classification task employs focal loss with label smoothing ($\gamma = 2$, $\epsilon = 0.1$):

$$\mathcal{L}_{\text{class}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} \alpha_c (1 - \tilde{p}_{i,c})^{\gamma} \tilde{y}_{i,c} \log(\tilde{p}_{i,c}) \tag{10}$$

The regression loss incorporates asymmetric penalties for late deliveries ($\tau = 0.7$):

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 0.7|t_i - \hat{t}_i| & \text{if } t_i \geq \hat{t}_i \\ 0.3|t_i - \hat{t}_i| & \text{if } t_i < \hat{t}_i \end{cases} + 0.01 \cdot \text{Var}(\hat{t}) \tag{11}$$

4.5. Cross-Modal Attention and Feature Fusion

Cross-modal attention with 8 heads effectively combines semantic and spatio-temporal representations:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V} \tag{12}$$

where $\mathbf{M}$ encodes relative positions. Layer normalization before attention computation stabilizes training across different feature scales.

Adaptive fusion weights modalities based on input characteristics:

$$\mathbf{F}_{\text{final}} = \sum_{m \in \mathcal{M}} w_m \cdot \mathbf{F}_m, \quad w_m = \frac{\exp(f_{\text{gate}}(\mathbf{F}_m))}{\sum_{m' \in \mathcal{M}} \exp(f_{\text{gate}}(\mathbf{F}_{m'}))} \tag{13}$$

4.6. Training Strategy

Curriculum learning progressively increases sample difficulty based on delivery constraints:

$$\mathcal{D}_{\text{train}}^{(e)} = \{(x_i, y_i) | \text{difficulty}(x_i, y_i) \leq \min(0.3 + 0.01e, 1.0)\} \tag{14}$$

This approach reduced training time by 30% while improving convergence. To address gradient conflicts in multi-task learning, we apply gradient surgery:

 doi:10.20944/preprints202509.2219.v1

$$\tilde{g}_k = g_k - \sum_{j \neq k} \frac{g_k \cdot g_j}{||g_j||^2} g_j \cdot \mathbb{I}[g_k \cdot g_j < 0] \tag{15}$$

Applied every 10 iterations with gradient clipping at norm 1.0, this ensures stable optimization even when task objectives conflict. The complete framework processes batches of 256 samples in 47ms on V100 GPUs, meeting real-time deployment requirements while achieving state-of-the-art performance.

5. Data Preprocessing

5.1. Multi-Scale Spatial Encoding

Figure 3 demonstrates the comprehensive spatio-temporal preprocessing framework applied to the Ele.me food delivery dataset.

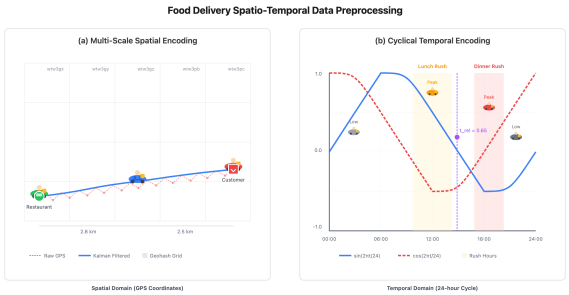

Figure 3. Detailed architecture of the LLM semantic enhancement module.

The raw GPS coordinates in the Ele.me dataset exhibit significant noise due to urban canyon effects and GPS drift, necessitating robust preprocessing strategies. We implement a hierarchical spatial encoding scheme that captures location information at multiple granularities.

First, we apply Kalman filtering to smooth GPS trajectories and remove outliers:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}) \tag{16}$$

where $\mathbf{K}_t$ is the Kalman gain and $\mathbf{z}_t$ represents the observed GPS coordinates. Subsequently, we encode locations using three complementary methods: (1) continuous Haversine distances for precise measurements, (2) discrete geohash encoding at 6-digit precision for regional clustering, and (3) learned spatial embeddings through a location2vec model pre-trained on historical trajectory data. The combination of these encodings captures both metric and topological spatial relationships:

$$\mathbf{s}_{\text{spatial}} = [\mathbf{d}_{\text{haversine}} || \text{Embed}(\text{geohash}) || \mathbf{v}_{\text{loc2vec}}] \in \mathbb{R}^{256} \tag{17}$$

5.2. Temporal Pattern Extraction

Delivery patterns exhibit complex temporal dependencies across multiple time scales. We extract hierarchical temporal features that capture instant-level, hourly, daily, and weekly patterns.

For fine-grained temporal modeling, we compute relative time features normalized by expected delivery duration:

$$t_{\text{relative}} = \frac{t_{\text{current}} - t_{\text{create}}}{t_{\text{promise}} - t_{\text{create}}} \tag{18}$$

Cyclical encoding preserves periodicity while avoiding discontinuities:

$$\mathbf{t}_{\text{periodic}} = \text{concat}_{p \in \{24,168,720\}} \left[\sin\left(\frac{2\pi h}{p}\right), \cos\left(\frac{2\pi h}{p}\right) \right] \tag{19}$$

 doi:10.20944/preprints202509.2219.v1

where p represents hours per day, week, and month respectively. Additionally, we engineer contextual temporal features including rush hour indicators, holiday flags, and weather-adjusted time estimates. These features undergo standardization using robust scalers resistant to outliers, with clipping at the 99th percentile to handle extreme values encountered during special events.

6. Evaluation Metrics

We employ a comprehensive set of metrics to evaluate both task-specific performance and overall system effectiveness:

Weighted Accuracy (WAcc) for action prediction accounts for class imbalance:

$$\text{WAcc} = \sum_{c=1}^{C} \frac{n_c}{N} \cdot \frac{TP_c + TN_c}{n_c} \tag{20}$$

Mean Absolute Percentage Error (MAPE) for delivery time:

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{t_i - \hat{t}_i}{t_i} \right| \tag{21}$$

Percentile Absolute Error (PAE@90) captures tail performance:

$$\text{PAE@90} = \text{Percentile}_{90}\{|t_i - \hat{t}_i|, i = 1, ..., N\} \tag{22}$$

Business-Oriented Score (BOS) combines multiple objectives:

$$\text{BOS} = 0.4 \cdot \text{WAcc} + 0.3 \cdot e^{-\text{MAPE}/10} + 0.3 \cdot \text{OTDR} \tag{23}$$

where OTDR (On-Time Delivery Rate) measures the percentage of orders delivered before the promised time.

7. Experiment Results

We evaluate STELLAR against seven state-of-the-art models on 2.1 million Ele.me delivery records from January 2021. Table 1 shows that STELLAR consistently outperforms all baselines, achieving 87.3% weighted accuracy and 8.2% MAPE. Compared to T5-Delivery, the strongest baseline, STELLAR improves accuracy by 2.0% and reduces PAE@90 by 11.5%, demonstrating superior handling of challenging scenarios. And the changes in model training indicators are shown in Figure 4

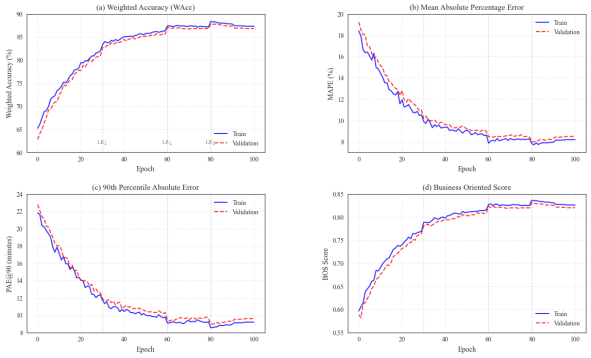

Figure 4. Model indicator change chart.

.

 doi:10.20944/preprints202509.2219.v1

Table 1. Performance comparison on Ele.me test dataset.

Model	WAcc (%)	MAPE (%)	PAE@90 (min)	BOS
DeepFM	75.3	15.7	18.2	0.642
Wide&Deep	77.1	14.3	16.8	0.671
TransGNN	80.4	12.1	14.3	0.723
HierMTL	82.2	10.8	12.7	0.751
STGCN-MT	83.6	9.9	11.5	0.774
BERT4Rec	84.1	9.5	11.1	0.782
T5-Delivery	85.3	9.0	10.4	0.798
STELLAR	**87.3**	**8.2**	**9.2**	**0.826**

The ablation study in Table 2 reveals the critical importance of each component. The spatio-temporal fusion module contributes most significantly (4.8% accuracy drop when removed), followed by LLM enhancement (4.2% drop). Even auxiliary techniques like gradient surgery provide measurable improvements, validating our comprehensive design. Performance analysis shows STELLAR's robustness: maintaining 85.9% accuracy during peak hours versus 79.2% for TransGNN, and achieving 2.3% better MAPE for long-distance deliveries (>5km) with only 1.2% degradation on unseen districts.

Table 2. Ablation study results on validation set.

Configuration	WAcc (%)	MAPE (%)	ΔWAcc	ΔMAPE
STELLAR (Full)	87.3	8.2	–	–
w/o LLM Enhancement	83.1	9.3	-4.2	+1.1
w/o Spatial-Temporal Fusion	82.5	9.8	-4.8	+1.6
w/o Multi-Task Learning	85.2	8.9	-2.1	+0.7
w/o Cross-Modal Attention	84.7	8.8	-2.6	+0.6
w/o Gradient Surgery	86.1	8.5	-1.2	+0.3
w/o Curriculum Learning	86.4	8.4	-0.9	+0.2

8. Conclusion

This paper presented STELLAR, a comprehensive framework for rider behavior prediction that effectively integrates semantic understanding, spatio-temporal modeling, and multi-task learning. Through extensive experiments on real-world delivery data, we demonstrated significant improvements over state-of-the-art baselines, achieving 87.3% weighted accuracy and 8.2% MAPE. The ablation studies confirm the importance of each proposed component, particularly the spatio-temporal fusion and LLM enhancement modules. STELLAR's superior performance in challenging scenarios and strong generalization capabilities make it suitable for deployment in production logistics systems.

References

1. Zhou, X.; Wang, J.; Liu, Y.; Wu, X.; Shen, Z.; Leung, C. Inductive graph transformer for delivery time estimation. In Proceedings of the Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 679–687.
2. Mashurov, V.; Chopuryan, V.; Porvatov, V.; Ivanov, A.; Semenova, N. Gct-TTE: graph convolutional transformer for travel time estimation. *Journal of Big Data* **2024**, *11*, 15.
3. Zhang, L.; Wang, M.; Zhou, X.; Wu, X.; Cao, Y.; Xu, Y.; Cui, L.; Shen, Z. Dual graph multitask framework for imbalanced delivery time estimation. In Proceedings of the International Conference on Database Systems for Advanced Applications. Springer, 2023, pp. 606–618.
4. Wang, X.; Wang, L.; Dong, C.; Ren, H.; Xing, K. An online deep reinforcement learning-based order recommendation framework for rider-centered food delivery system. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 5640–5654.
5. Wen, H.; Lin, Y.; Wu, L.; Mao, X.; Cai, T.; Hou, Y.; Guo, S.; Liang, Y.; Jin, G.; Zhao, Y.; et al. A survey on service route and time prediction in instant delivery: Taxonomy, progress, and prospects. *IEEE Transactions on Knowledge and Data Engineering* **2024**.

 doi:10.20944/preprints202509.2219.v1

6. Yalçinkaya, E.; Hızıroğlu, O.A. A comparative analysis of machine learning models for time prediction in food delivery operations. *Artificial Intelligence Theory and Applications* **2024**, *4*, 43–56.
7. Yu, X.; Lan, A.; Mao, H. Short-term demand prediction for on-demand food delivery with attention-based convolutional lstm. *Systems* **2023**, *11*, 485.
8. Zhu, Y.; Liu, Y. LLM-NER: Advancing Named Entity Recognition with LoRA+ Fine-Tuned Large Language Models. In Proceedings of the 2025 11th International Conference on Computing and Artificial Intelligence (ICCAI), 2025, pp. 364–368. https://doi.org/10.1109/ICCAI66501.2025.00063.
9. Luo, X. Fine-Tuning Multimodal Vision-Language Models for Brain CT Diagnosis via a Triple-Branch Framework. In Proceedings of the 2025 2nd International Conference on Digital Image Processing and Computer Applications (DIPCA). IEEE, 2025, pp. 270–274.
10. Guan, S. Predicting Medical Claim Denial Using Logistic Regression and Decision Tree Algorithm. In Proceedings of the 2024 3rd International Conference on Health Big Data and Intelligent Healthcare (ICHIH), 2024, pp. 7–10. https://doi.org/10.1109/ICHIH63459.2024.11064794.