Abstract

Real-time dispatching is a fundamental problem in on-demand delivery systems. However, most existing solutions treat travel time prediction and dispatch optimization as separate modules, limiting their ability to respond effectively to dynamic traffic conditions, stochastic demand, and operational uncertainty. Few studies integrate Graph Neural Network (GNN)-based ETA learning with Deep Reinforcement Learning (DRL) for spatiotemporal dispatch decision-making. To address this gap, this thesis proposes an ETA-aware dispatch framework for learning to dispatch in on-demand delivery. Specifically, the dispatch process is formulated as a constrained Markov decision process, where feasible rider–order assignments are generated through route insertion under precedence and capacity constraints. Within this framework, a GNN-based ETA module provides predictive travel time signals and uncertainty estimates, which are incorporated into three dispatch strategies: Greedy selection, Pareto-based multi-objective filtering, and MaskablePPO-based reinforcement learning over dynamically constrained action spaces. A unified evaluation pipeline is further developed to assess effectiveness, runtime consistency, robustness under noise, and the effect of removing risk terms. Experiments across multiple environment scales show that Pareto-based filtering consistently outperforms Greedy dispatch in cumulative reward, while the DRL-based strategy remains unstable in larger constrained settings. The results suggest that integrating GNN-based prediction and DRL-based optimization is a promising direction for spatiotemporal decision-making in intelligent delivery systems, although stable learning under uncertainty remains a major challenge.

KEYWORDS | ON-DEMAND DELIVERY, GRAPH NEURAL NETWORKS, DEEP REINFORCEMENT LEARNING, ETA PREDICTION

1. Introduction

The explosive growth of on-demand delivery platforms, has introduced new challenges in urban logistics, particularly regarding real-time order assignment and travel time prediction under dynamic traffic and demand conditions. These platforms must continuously assign incoming orders to couriers, considering evolving factors such as road congestion, fluctuating demand density, rider availability, and customer deadlines. Traditional approaches based on rule-based heuristics or static optimization models often fall short in these environments, as they typically assume stationarity or re-optimize in batch without accounting for sequential dependencies.

Graph Neural Networks (GNNs) and Deep Reinforcement Learning (DRL) have recently emerged as promising techniques to tackle the core sub-problems in intelligent delivery systems. GNNs are well-suited for modeling delivery data, which is inherently structured as spatial-temporal graphs—comprising locations (e.g., restaurants, riders, and customers) and dynamic links (e.g., traffic-aware routes). Recent studies such as KDG-ETA (Zhang et al., 2024) and Graph2RETA (Sentanoe et al., 2024) demonstrate that GNN-based models can significantly improve Estimated Time of Arrival (ETA) predictions by capturing complex dependencies in trajectory and traffic data. These methods integrate static road features with dynamic conditions and learn transferable representations across regions and delivery scenarios.

Meanwhile, dispatching decisions are inherently sequential and require continuous adaptation, making DRL a natural fit. DRL agents learn to optimize long-term objectives by interacting with the environment and adjusting policies based on real-time feedback. Notable applications, including TCAC-Dispatch (Guo et al., 2021), illustrate how DRL can outperform greedy or myopic algorithms in both efficiency and service quality by considering the long-term impact of assignment decisions and supporting concurrent delivery with overlapping tasks. In addition, DRL methods have shown the capability to address workload imbalance among couriers, a growing concern in platform sustainability, by learning equitable dispatching behaviors.

Despite the individual successes of GNNs in prediction and DRL in control, their integration remains underexplored, particularly in real-world, dynamic dispatch environments. Most current systems treat prediction and decision-making as separate modules, which limits the ability to reason holistically about uncertainty and optimization. However, recent works such as Chen et al. (2024) and Gebreyesus et al. (2025) point toward a new direction: embedding graph-based state representations directly into the DRL decision pipeline, enabling joint learning of structured spatial-temporal features and adaptive dispatch strategies.

This study aims to systematically investigate how the integration of GNN-based Estimated Time of Arrival (ETA) prediction with DRL-based dispatching can enhance the performance of on-demand delivery platforms under volatile traffic and demand conditions. It further explores how this integration supports multi-objective optimization, balancing operational efficiency with rider workload fairness. It also seeks to clarify the extent to which such an integrated framework can improve the robustness and adaptability of real-time dispatching in complex spatiotemporal settings.

2. Literature Review

Graph Neural Networks (GNNs) have proven highly effective for modeling the spatial and temporal dependencies inherent in delivery systems and urban mobility. By treating road networks or delivery routes as graphs, GNN-based models naturally capture how travel times depend on both network structure and dynamic traffic conditions. Wang et al. (2021) introduce GraphTTE, an attention-enhanced spatiotemporal GNN that integrates static road network features with dynamic traffic states (via Graph Convolutional layers and GRU units) to predict travel times. Jin et al. (2024) propose a Spatio-Temporal Dual GNN (STDGNN) that models a road network with two coupled graphs – one for intersections and one for road segments. This dual-graph approach captures multi-scale spatial-temporal correlations and yields more accurate travel time estimates than prior methods.

Beyond general traffic networks, GNNs have been tailored to last-mile delivery and logistics contexts. Zhang et al. (2024) develop a knowledge-distillation GNN for package delivery time prediction (KDG-ETA) that embeds historical trajectory knowledge into origin-destination node representations. In the on-demand food delivery domain, Sentanoe et al. (2024) present Graph2RETA, a dynamic spatial-temporal GNN that simultaneously predicts couriers' future routes and arrival times. Likewise, Betkier (2025) reports that a GCN-based model incorporating road attributes and conditions (via a graph of the road network combined with an MLP for features) can predict travel times with only ~8% MAPE, markedly better than conventional neural nets. Additional graph-based approaches have shown strong generalization in various urban delivery contexts (Wan et al., 2022; Zhao et al., 2023; Lan et al., 2022), and new hybrids combining GNNs and Transformer-like attention layers further boost performance across long-range delivery tasks.

There are many studies applying machine learning approaches for estimating travel time. Machine learning tools have demonstrated strong capabilities in modeling delivery systems by learning from vast historical data without the need for handcrafted rules. In travel time estimation, models such as support vector regression, random forests, and gradient boosting have shown competitive accuracy in short-term urban mobility prediction (Zhang & Haghani, 2015; Wu et al., 2015). In last-mile logistics, traditional machine learning approaches like KNN, SVR, and boosting were widely used before the adoption of graph-based models.

Moreover, deep Reinforcement Learning (DRL) has emerged as a key methodology for adaptive dispatching under uncertain, high-frequency delivery environments. Traditional rule-based dispatch heuristics are limited in coping with fluctuating demand, traffic variability, and spatiotemporal dynamics. DRL addresses these limitations by learning state-action mappings

to optimize long-term efficiency and adaptability. Wang et al. (2023) proposed a time-constrained actor‑critic method tailored for multi-order assignments in food delivery, while Liu et al. (2022) developed a deep dispatching policy to adaptively match supply-demand imbalance in ride-hailing. Huang et al. (2021) extended DRL to UAV-based delivery with stochastic scheduling constraints, and Silva et al. (2023) demonstrated DRL’ s robustness under uncertain crowdsourced logistics environments. These works show that DRL is broadly applicable across various dispatch scenarios including batching, load balancing, and dynamic agent coordination.

Despite these advances, dispatching and ETA forecasting are often treated as modular components. Few studies have fully integrated predictive models into decision policies. Gebreyesus et al. (2025) proposed embedding spatiotemporal GNN outputs into DRL scheduling agents, while Chen et al. (2024) and Li et al. (2021) applied graph-based encoders to represent courier-task relationships. Although promising, these approaches are often restricted to static settings or partial integration. A fully unified GNN-DRL framework optimized for real-time dispatch under uncertainty remains an open and impactful research frontier.

3. Methodology

3.1 Problem formulation

Online dispatching in on-demand delivery is modeled over a directed transportation graph $G = (\mathcal{V}, \mathcal{E})$. A travel-time query on an edge $e \in \mathcal{E}$ is time-dependent and stochastic, denoted by a nonnegative random variable $\tau_e(t)$ parameterized by departure time t. Time-dependent shortest-path primitives motivate the ETA subroutine used by the dispatch kernel, and establish the principled role of a shortest-ETA operator when network delays vary over time.

Orders arrive online and must be assigned to riders under strict runtime constraints. Such problems inherit the well-known combinatorial nature and NP-hardness of dynamic pickup-and-delivery settings, especially under insertion-based routing and online arrivals (Berbeglia et al., 2010; Liang et al., 2024).

The dispatch control process is cast as a constrained Markov decision process (CMDP)

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{C}),$$

with state $s_t \in \mathcal{S}$, action $a_t \in \mathcal{A}$, transition kernel $\mathcal{P}$, reward $\mathcal{R}$, discount $\gamma \in (0,1]$, and constraint set $\mathcal{C}$ defining feasibility. At decision step t, the system state is

$$s_t = \left(t,\ \mathcal{O}_t,\ \mathcal{R}_t,\ \{\pi_r^t\}_{r\in\mathcal{R}_t},\ \Theta_t\right),$$

where $\mathcal{O}_t$ is the set of pending orders, $\mathcal{R}_t$ is the set of riders, π_r^t is the planned stop sequence for rider r, and Θ_t denotes all auxiliary features required by the policy and the insertion kernel. Each order $o \in \mathcal{O}_t$ is represented as

$$o = (p_o,\ d_o,\ a_o,\ \ell_o,\ \text{meta}_o),$$

with pickup node $p_o \in \mathcal{V}$, dropoff node $d_o \in \mathcal{V}$, arrival time a_o, load ℓ_o, and metadata meta_o. Each rider $r \in \mathcal{R}_t$ is represented as

$$r = (x_r,\ c_r,\ \pi_r^t,\ \text{meta}_r),$$

with current location x_r, capacity c_r, route π_r^t, and optional metadata.

Dispatching uses route insertion, for a rider r with current route $\pi_r^t = (v_0, v_1, \ldots, v_L)$, an action assigns an order o and insertion indices (i, j) such that pickup precedes dropoff

$$a_t = (r, o, i, j), \quad 1 \le i < j \le L + 2.$$

The resulting route is

$$\pi_r^{t+} = \text{Insert}(\pi_r^t,\ p_o@i,\ d_o@j),$$

where insertion places p_o at position i and d_o at position j after accounting for index shifts. A null action $a_t = \varnothing$ can represent "no dispatch" when no feasible insertion exists or when rejection is allowed. The constraints define a feasible set

$$\mathcal{A}_{\text{feas}}(s_t) = \{a \in \mathcal{A}:\ \pi_r^{t+} \text{ satisfies all constraints in } \mathcal{C}\}.$$

Given a feasible action a_t, the transition updates the selected rider route, advances simulation time, realizes travel-time random variables, and adds new online arrivals:

$$s_{t+1} \sim \mathcal{P}(\cdot|\ s_t, a_t).$$

Because online arrivals and realized travel times are stochastic, $\mathcal{P}$ is non-deterministic even under deterministic policy evaluation. This structure follows standard sequential decision modeling in real-time dispatch systems (Liang et al., 2024; Dai et al., 2026).

A decomposed immediate reward is designed to couple operational objectives with ETA-related regularization

$$r_t = \mathcal{R}(s_t, a_t) = u_{\text{service}}(s_t, a_t)\ -\ \lambda_{\text{delay}}\, g_{\text{delay}}(s_t, a_t)\ -\ \alpha\, \Delta\mu(s_t, a_t)\ -\ \beta\, \Delta\sigma(s_t, a_t),$$

where u_{service} captures service completion and efficiency, g_{delay} penalizes tardiness or SLA violations, and $\Delta\mu, \Delta\sigma$ are incremental ETA mean and uncertainty costs induced by a route insertion.

3.2 Route Insertion and Cost Evaluation

The backbone dispatch kernel enumerates feasible insertions, evaluates incremental costs, and returns a candidate set for greedy, Pareto, or RL selection. Let a route be a stop sequence $\pi = (v_0, \dots, v_L)$, with v_0 denoting current rider location (or a virtual start node). A generic additive route cost is defined as

$$C(\pi; t_0) = \sum_{k=0}^{L-1} c\,(v_k, v_{k+1};\ \phi_k)\ +\ \Psi(\pi),$$

where $c(\cdot)$ is the segment cost (based on ETA predictions or shortest-path queries), ϕ_k denotes the context for segment k (departure time, temporal features, or other conditioning variables), and $\Psi(\pi)$ is a constraint penalty term. Under risk-aware ETA, the segment cost is

$$c(u, v; \phi) = \mu(u, v; \phi) + \beta\,\sigma(u, v; \phi),$$

and $\Psi(\pi)$ includes feasibility-related terms when soft constraints exist. Using μ and σ to form a risk cost is consistent with uncertainty-aware optimization and robust planning practice when reliability matters (Kendall & Gal, 2017; Mallick et al., 2024; Tang et al., 2025).

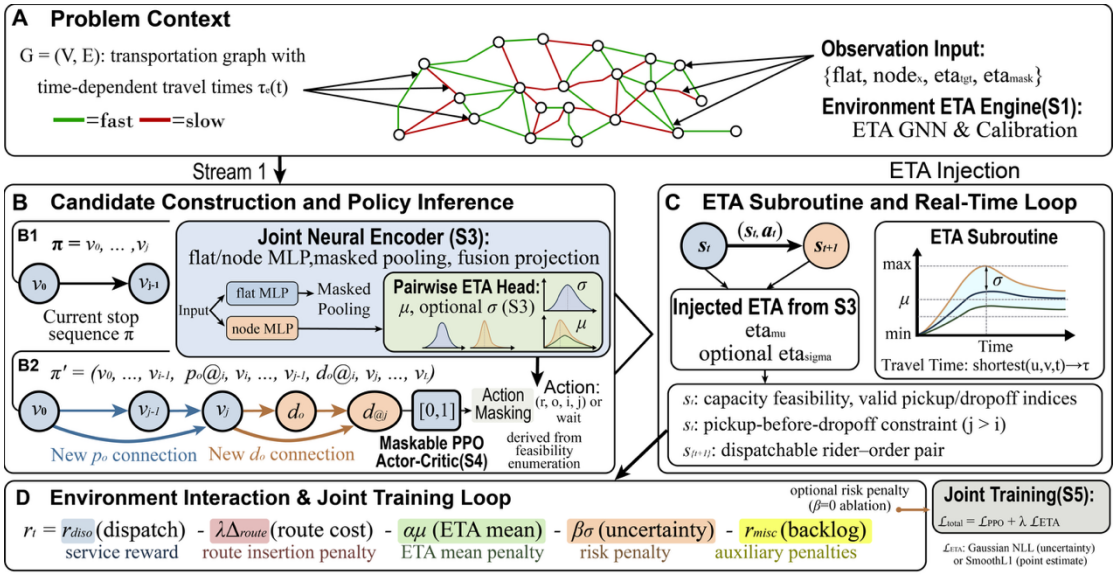

Figure 1. Architecture of the proposed ETA-aware dispatch framework, which consists of four components: candidate construction, joint neural encoding and policy inference, ETA subroutine with real-time feedback, and environment interaction with reward decomposition.

Given a rider route $\pi = (v_0, \dots, v_L)$ and a new order with pickup p and dropoff d, consider insertion indices $i < j$. Denote the post-insertion route by $\pi' = \mathrm{Insert}(\pi, p@i, d@j)$. The insertion delta is

$$\Delta C(i,j) = C(\pi'; t_0) - C(\pi; t_0).$$

Feasibility is verified by forward simulation over the route. For route index k, define onboard load q_k and stop type. Precedence and capacity constraints are

$$\forall o\colon \mathrm{pos}(p_o) < \mathrm{pos}(d_o), \qquad \forall k\colon q_k \le c_r.$$

Travel time between nodes is estimated via a time-dependent shortest-path function shortest eta(u, v, t), which returns a travel-time estimate between nodes u and v at departure time t. A principled implementation is a time-dependent shortest-path computation:

$$\eta(u,v,t) = \min_{\mathcal{P}:u\to v} \sum_{e\in\mathcal{P}} \tau_e\,(t_e),$$

with FIFO or waiting constraints depending on the environment. Time-dependent shortest-path algorithms and their properties justify this operator's role in a dispatch kernel that must react to time-varying conditions (Orda & Rom, 1990).

3.3 ETA learning with uncertainty

An ETA model f_θ maps an input representation z to a mean and an uncertainty scale:

$$\big(\mu_\theta(z), \sigma_\theta(z)\big) = f_\theta(z), \qquad \sigma_\theta(z) > 0.$$

The scale σ models heteroscedastic aleatoric uncertainty. Epistemic uncertainty is modeled by an ensemble $\{f_{\theta_m}\}_{m=1}^{M}$ (or other Bayesian approximations), producing

$$\mu(z) = \frac{1}{M}\sum_{m=1}^{M} \mu_{\theta_m}(z), \qquad \mathrm{Var}_{\mathrm{epi}}(z) = \frac{1}{M}\sum_{m=1}^{M} (\mu_{\theta_m}(z) - \mu(z))^2,$$

and total predictive variance

$$\mathrm{Var}_{\mathrm{tot}}(z) = \underbrace{\frac{1}{M}\sum_{m=1}^{M} \sigma_{\theta_m}^2(z)}_{\mathrm{Var}_{\mathrm{ale}}(z)} + \underbrace{\frac{1}{M}\sum_{m=1}^{M} \mu_{\theta_m}^2(z) - \left(\frac{1}{M}\sum_{m=1}^{M} \mu_{\theta_m}(z)\right)^2}_{\mathrm{Var}_{\mathrm{epi}}(z)}$$

This decomposition follows established uncertainty taxonomy and deep ensemble practice (Kendall & Gal, 2017; Lakshminarayanan et al., 2017; Mallick et al., 2024).

A road-network-aware ETA model is naturally expressed via message passing on graphs. A generic GNN layer updates node embeddings:

$$h_v^{(\ell+1)} = \phi\left(h_v^{(\ell)},\ \mathrm{AGG}\left\{\psi\left(h_v^{(\ell)}, h_u^{(\ell)}, e_{uv}\right) : (u,v) \in \mathcal{E}\right\}\right),$$

with ϕ, ψ learnable functions and AGG a permutation-invariant aggregator. Graph-based ETA and spatiotemporal inference are widely adopted in intelligent transportation modeling due to structured spatial dependence (Chen et al., 2024; Fang et al., 2024; Mallick et al., 2024).

Input features typically include node location, zone identifiers, time-of-day encodings, historical traffic aggregates, and route context features. Exact feature definitions, depth, hidden dimensions.

Let y be observed travel time. Under a Gaussian likelihood

$$y \mid z \sim \mathcal{N}(\mu_\theta(z), \sigma_\theta(z)^2),$$

the negative log-likelihood loss is

$$\mathcal{L}_{\text{NLL}}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{(y_n - \mu_\theta(z_n))^2}{2\sigma_\theta(z_n)^2} + \frac{1}{2}\log(\sigma_\theta(z_n)^2)\right] + \text{const.}$$

This is a standard way to learn data-dependent uncertainty and connect predictive calibration to likelihood-based scoring (Kendall & Gal, 2017; Mallick et al., 2024).

Calibration is evaluated using proper scoring rules and interval-coverage diagnostics. For point accuracy, MAE and RMSE are standard:

$$\text{MAE} = \frac{1}{N}\sum_{n} |y_n - \mu_n|, \qquad \text{RMSE} = \sqrt{\frac{1}{N}\sum_{n} (y_n - \mu_n)^2}.$$

3.4 Decision Strategies

The dispatch score uses a risk-adjusted ETA cost:

$$\text{RiskCost}(z) = \mu(z) + \beta\sigma(z), \qquad \beta \geq 0.$$

The action space is a time-varying discrete set of insertion candidates $\mathcal{C}_t \subseteq \mathcal{A}_{\text{feas}}(s_t)$, produced by the insertion kernel. Because $|\mathcal{C}_t|$ is variable, the policy uses padded logits and a binary mask $m_t \in \{0,1\}^{A_{\max}}$ where $m_t[a] = 1$ if action index a corresponds to a valid candidate. The masked policy is

$$\pi_\theta(a \mid s_t, m_t) = \frac{\exp\big(\ell_\theta(a \mid s_t)\big)\, m_t[a]}{\sum_{a'} \exp\big(\ell_\theta(a' \mid s_t)\big)\, m_t[a']},$$

which assigns zero probability to infeasible actions by construction. Theoretical and empirical analysis of invalid-action masking supports its use when invalid action density is high (Huang & Ontañón, 2020).

A PPO-style clipped objective is adopted:

$$\mathcal{L}_{\text{clip}}(\theta) = \mathbb{E}\big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\big],$$

where $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$, ϵ is the clip parameter, and $\hat{A}_t$ is an advantage estimator. Peer-reviewed T-ITS evidence demonstrates PPO's practicality in large state spaces and constrained decision settings, including systems that explicitly combine PPO with pruning rules for scalability (Staffolani et al., 2025).

Stable open-source implementations are documented and benchmarked in JMLR (Raffin et al., 2021).

The advantage estimator is computed via a generalized temporal-difference residual expansion:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l\, \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\varphi(s_{t+1}) - V_\varphi(s_t),$$

with $\lambda \in [0,1]$. Recent IEEE Transactions on Games work analyzes advantage estimation variants for PPO and the bias-variance trade-off under truncated rollouts (Jin et al., 2024).

Each candidate insertion $c \in \mathcal{C}_t$ is associated with a multi-objective vector

$$g(c) = \big(\Delta\text{Utility}(c),\ \Delta\mu(c),\ \Delta\sigma(c)\big).$$

A candidate c Pareto-dominates c' if it is no worse in all components and strictly better in at least one. Non-dominated filtering yields a Pareto set $\mathcal{P}_t$, followed by tie-breaking via a scalar score (for example, the reward-shaped score used in the kernel). Pareto-front methods are canonical in multi-objective optimization and provide a transparent baseline when objectives are conflicting (Deb et al., 2002; Roijers et al., 2013).

3.5 Algorithms

Algorithm 1. Masked Route Insertion Dispatch

Input: s_t, C_r, C_o, Q, K, $\hat{E}$, $\hat{\Sigma}$, α, β
Output: a_t, r_t, s_{t+1}
$(C_r, C_o) \leftarrow \text{GetCandidates}(s_t)$;
$\mathcal{F} \leftarrow \varnothing$;
for $r_i \in C_r$ **do**
  for $o_j \in C_o$ **do**
    for $0 \le p \le |Q_i|,\ p+1 \le d \le |Q_i|+1$ **do**
      $Q_i' \leftarrow \text{Insert}(Q_i, o_j, p, d)$;
      if $\text{Feasible}(Q_i')$ **then**
        $\Delta_{ijpd} \leftarrow \text{Cost}(Q_i') - \text{Cost}(Q_i)$;
        $\mathcal{F} \leftarrow \text{TopK}(\mathcal{F} \cup \{(\Delta_{ijpd}, i, j, p, d)\}, K)$;
$M \leftarrow \Pi(\mathcal{F})$;
$a_t \sim \pi_\theta(\cdot \mid s_t, M)$;
if $a_t = \text{wait}$ **then**
  $r_t \leftarrow r_{\text{wait}}$;
else if $\neg\text{DispatchValid}(a_t)$ **then**
  $r_t \leftarrow r_{\text{inv}}$;

else
  $\mu_{ij} \leftarrow \mathrm{clip}(\hat{E}_{ij}, 0.5E_{ij}^{fb}, 1.5E_{ij}^{fb})$;
  $\sigma_{ij} \leftarrow \hat{\Sigma}_{ij}$;
  $\mathrm{Dispatch}(r_i, o_j, p, d, \mu_{ij})$;
  $r_t \leftarrow 1 - 0.01\Delta_{ijpd} - \alpha\mu_{ij} - \beta\sigma_{ij}$;
  $r_t \leftarrow r_t + b_{\mathrm{disp}} - b_{\mathrm{rep}} - b_{\mathrm{backlog}}$;
$s_{t+1} \leftarrow \mathrm{Advance}(s_t)$;

Algorithm 2 Training and Evaluation Workflow Used in the Repository

Input: c, T, H, N, S, λ_η
Output: π_θ, Φ
$D \leftarrow \mathrm{BuildDataset}(c)$;
$\psi \leftarrow \mathrm{TrainEtaGNN}(D_{\mathrm{edge}})$;
$\theta \leftarrow \mathrm{InitEncoder}()$;
$\pi_\theta \leftarrow \mathrm{InitJointMaskablePPO}(\theta)$;
while *collected steps* $< T$ **do**
  $\mathcal{B} \leftarrow \varnothing$;
  for $t = 1, \ldots, H$ **do**
    $o_t = \{x_t, n_t, \eta_t, m_t\} \leftarrow \mathrm{Env}(D, \psi)$;
    $M_t \leftarrow \mathrm{ActionMask}(o_t)$;
    $(\hat{E}_t, \hat{\Sigma}_t) \leftarrow f_\theta(o_t)$;
    $a_t \sim \pi_\theta(\cdot \mid o_t, M_t)$;
    $\mathrm{Inject}(\hat{E}_t, \hat{\Sigma}_t)$;
    $(o_{t+1}, r_t) \leftarrow \mathrm{EnvStep}(a_t)$;
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{(o_t, a_t, r_t, o_{t+1}, M_t)\}$;
  $\theta \leftarrow \arg\min_\theta \left(\mathcal{L}_{\mathrm{PPO}} + \lambda_\eta \mathcal{L}_{\mathrm{ETA}}\right)$;
foreach $s \in \{S_1, \ldots, S_N\}$ **do**
  $\mathrm{Reset}(D, s)$;
  $a_t \leftarrow \mathrm{PolicyOrBaseline}$;
  $a_t \leftarrow \mathrm{Fallback}(a_t) \in \{\mathrm{greedy}, \eta\text{-greedy}, \mathrm{wait}\}$;
  $(R_s, \mathrm{MAE}_s, \mathrm{RMSE}_s, \mathrm{NLL}_s) \leftarrow \mathrm{RolloutEval}(s)$;
$\Phi \leftarrow \mathrm{Aggregate}\big(\{R_s, \mathrm{MAE}_s, \mathrm{RMSE}_s, \mathrm{NLL}_s\}_{s=1}^{N}\big)$;
$\mathrm{Export}(\Phi)$;

4. Experiments

4.1 Experimental Setup

The evaluation is organized into five experiment groups designed to validate effectiveness, scalability, robustness under noise, ablation of risk terms, and non-triviality via a random baseline. The experiments are deterministic at policy execution time, and seeds are explicitly controlled. This evaluation style reduces stochastic variance and enables paired testing across methods under the same seeds (Raffin et al., 2021; Staffolani et al., 2025).

For contextual positioning, industrial-scale dispatch systems typically operate in dynamic and uncertain environments and solve NP-hard assignment structures within seconds (Liang et al., 2024). Meanwhile, academic benchmarks often use grid or region discretization for controlled study and scalability analysis (Jing et al., 2025; Wang et al., 2024).

Three scenario scales are evaluated: small, main, and large. The presence of multiple scales is essential because insertion-based candidate enumeration and shortest-ETA computations can

exhibit nonlinear computational growth and accuracy degradation with scale, and this pattern is emphasized both in practical dispatch systems and in recent research on large-scale dispatching and uncertainty (Liang et al., 2024; Chen et al., 2024; Mallick et al., 2024). The evaluated scenarios are generated from environment configuration files and input datasets (orders, riders, nodes, and travel-time tables).

4.2 Baselines and compared methods

The experimental suite evaluates five methods: Greedy, Pareto, UETA (MaskablePPO), Random, and an Ablation variant without risk terms.

A: Effectiveness in the final setting

The final setting evaluates all methods under a consistent simulation protocol using multiple random seeds. Performance is measured by reward, ETA MAE, ETA RMSE, and ETA NLL.

B: Consistency under runtime conditions

A short-horizon consistency evaluation is conducted under the runtime setting to assess stability across repeated runs.

C: Robustness under noise perturbations

Robustness is evaluated under multiple noise levels (low, medium, high) applied to travel-time estimates. Specifically, the perturbed travel time is defined as

$$\tilde{\tau} = \tau \cdot (1 + \epsilon), \quad \epsilon \sim \mathcal{D}(0, \delta),$$

with δ taking three levels. Concrete δ values are unspecified and must match the simulator configuration.

D: Ablation without risk terms

An ablation setting disables ETA-based risk terms to isolate the contribution of risk-aware regularization.

E: Random baseline comparison

A random baseline is included as a reference to assess the non-triviality of structured dispatch strategies.

Metrics:

Performance is evaluated using both operational and predictive criteria.

Operational performance is measured by cumulative reward (higher is better), while

ETA estimation accuracy is quantified using MAE, RMSE, and NLL (lower is better).

Results are reported as averages over multiple independent runs with fixed random seeds.

4.3 Results

This section presents a comparison of dispatch strategies, including Greedy, Pareto-based selection, and the proposed risk-aware reinforcement learning approach (UETA), along with Random and Ablation baselines. Experiments are conducted across three environment scales under multiple evaluation settings. Performance is assessed using both operational metrics (reward) and ETA prediction metrics (MAE, RMSE, and NLL).

Table I reports the final setting results aggregated as mean ± standard deviation.

Table I. Overall performance in the final setting (mean ± std).

Scale	Method	Reward (↑)	ETA MAE (↓)	ETA RMSE (↓)	ETA NLL (↓)$^{(\times 10^5)}$
Small	Greedy	3.414 ± 0.140	9.174 ± 1.863	9.174 ± 1.863	3.538 ± 0.450
	Pareto	5.280 ± 0.097	9.994 ± 1.377	12.334 ± 1.451	6.060 ± 2.752
	UETA (MaskablePPO)	0.836 ± 0.574	9.994 ± 1.377	12.334 ± 1.451	6.060 ± 2.752
	Random	2.832 ± 0.277	N/A	N/A	N/A
	Ablation (α=0, β=0)	5.492 ± 0.080	10.137 ± 1.430	12.536 ± 1.521	6.505 ± 2.879
Main	Greedy	8.340 ± 0.125	18.538 ± 1.129	18.538 ± 1.129	0.131 ± 0.003
	Pareto	18.796 ± 0.127	21.328 ± 0.739	24.566 ± 0.773	3.320 ± 1.077
	UETA (MaskablePPO)	-10.328 ± 0.519	21.328 ± 0.739	24.566 ± 0.773	3.320 ± 1.077
	Random	6.080 ± 1.503	N/A	N/A	N/A
	Ablation (α=0, β=0)	20.656 ± 0.124	21.275 ± 0.626	24.484 ± 0.668	3.238 ± 1.026
Large	Greedy	9.493 ± 0.167	25.068 ± 1.277	25.068 ± 1.277	1.369 ± 0.031
	Pareto	27.315 ± 0.326	26.385 ± 0.656	29.525 ± 0.605	1.554 ± 0.517
	UETA (MaskablePPO)	-19.768 ± 0.595	26.385 ± 0.656	29.525 ± 0.605	1.554 ± 0.517
	Random	8.037 ± 1.837	N/A	N/A	N/A
	Ablation (α=0, β=0)	30.041 ± 0.338	26.434 ± 0.656	29.594 ± 0.606	1.610 ± 0.574

Pareto consistently outperforms Greedy in terms of reward across all scales, with relative improvements of 54.6% (Small), 125.4% (Main), and 187.7% (Large). This highlights the effectiveness of multi-objective candidate filtering in insertion-based dispatch, where trade-offs between delay and uncertainty are critical (Roijers et al., 2013; Liang et al., 2024; Chen et al., 2024).

UETA (MaskablePPO) underperforms in the Main and Large settings, yielding negative mean rewards. This suggests that the current policy learning setup is sensitive to reward scaling and constrained action spaces, highlighting the challenge of stable reinforcement learning in combinatorial dispatch problems (Huang and Ontañón, 2022; Staffolani et al., 2025; Wang et al., 2024).

Random performs significantly worse than structured methods, confirming the non-trivial nature of the dispatch task.

For Greedy, ETA RMSE equals ETA MAE in the exported results, indicating that both metrics are derived from the same aggregated values. Therefore, RMSE should not be interpreted as an independent metric in this case. ETA metrics for Random are reported as N/A due to missing ETA-related signals in the evaluation pipeline.

Table II. Performance in the runtime setting (mean ± std).

Scale	Method	Reward (↑)	ETA MAE (↓)	ETA RMSE (↓)	ETA NLL (↓)$^{(\times 10^5)}$
Small	Greedy	3.384 ± 0.150	9.631 ± 1.951	9.631 ± 1.951	3.538 ± 0.404
	Pareto	5.258 ± 0.100	10.248 ± 1.467	12.741 ± 1.644	6.594 ± 3.010
	UETA (MaskablePPO)	0.734 ± 0.587	10.248 ± 1.467	12.741 ± 1.644	6.594 ± 3.010
Main	Greedy	8.376 ± 0.140	18.401 ± 1.145	18.401 ± 1.145	0.131 ± 0.004
	Pareto	18.778 ± 0.150	21.371 ± 0.603	24.563 ± 0.619	3.259 ± 0.996
	UETA (MaskablePPO)	-10.501 ± 0.293	21.371 ± 0.603	24.563 ± 0.619	3.259 ± 0.996
Large	Greedy	9.450 ± 0.178	24.941 ± 1.189	24.941 ± 1.189	1.374 ± 0.028
	Pareto	27.184 ± 0.305	26.631 ± 0.707	29.711 ± 0.535	1.684 ± 0.636
	UETA (MaskablePPO)	-19.884 ± 0.549	26.645 ± 0.728	29.713 ± 0.537	1.684 ± 0.636

Table III. Robustness under noise (mean ± std).

Noise	Method	Reward (↑)	ETA MAE (↓)	ETA RMSE (↓)	ETA NLL (↓)$^{(\times 10^5)}$
Low	Greedy	19.781 ± 0.156	19.285 ± 0.324	19.285 ± 0.324	0.0056 ± 0.0002
	Pareto	25.006 ± 0.216	16.314 ± 0.434	18.342 ± 0.374	0.1037 ± 0.0440
	UETA (MaskablePPO)	-4.022 ± 0.720	16.318 ± 0.444	18.349 ± 0.377	0.1038 ± 0.0440
Mid	Greedy	19.782 ± 0.156	19.284 ± 0.324	19.284 ± 0.324	0.0056 ± 0.0001
	Pareto	25.007 ± 0.215	16.314 ± 0.435	18.342 ± 0.375	0.1037 ± 0.0440
	UETA (MaskablePPO)	-4.021 ± 0.719	16.318 ± 0.443	18.349 ± 0.377	0.1037 ± 0.0440

High	Greedy	19.782 ± 0.156	19.285 ± 0.304	19.285 ± 0.304	0.0056 ± 0.0001
	Pareto	25.007 ± 0.215	16.313 ± 0.435	18.341 ± 0.374	0.1036 ± 0.0440
	UETA (MaskablePPO)	-4.020 ± 0.719	16.317 ± 0.442	18.348 ± 0.376	0.1036 ± 0.0440

Robustness analysis shows that reward remains largely stable across low, medium, and high noise levels (Table III). This suggests that the current noise injection does not substantially alter the relative ordering of candidate actions under the tested settings. As a result, the observed robustness should be interpreted within the scope of the implemented perturbation levels.

In the runtime setting (Table II), Pareto consistently outperforms Greedy across all scales, while UETA exhibits unstable performance with negative rewards in larger environments. These trends are consistent with those observed in the final setting, indicating that multi-objective filtering remains effective under short-horizon evaluation.

Pairwise comparisons on reward indicate statistically significant differences between methods evaluated under identical seed settings (Table I). In particular, Pareto consistently outperforms Greedy across all scales, while UETA shows significantly different performance patterns. All reported p-values are below 0.001, indicating statistically significant differences across methods.

The ablation results (Table I) show that removing ETA-based risk terms leads to improved reward across all scales, suggesting that the current (α, β) configuration may impose overly strong penalties relative to operational rewards. This highlights the importance of careful calibration of risk-aware objectives in dispatch optimization (Bertsimas and Sim, 2004; Tamar et al., 2015; Tang et al., 2025).

Overall, the results demonstrate that multi-objective candidate filtering provides consistent improvements over greedy dispatch, while reinforcement learning approaches remain sensitive to reward design and action constraints. These findings highlight both the effectiveness of structured decision rules and the challenges of integrating uncertainty-aware learning into large-scale dispatch systems.

From a computational perspective, the dominant cost arises from candidate enumeration and ETA queries, which scale as $O(|\mathcal{O}_t|\,|\mathcal{R}_t|\,L^2)$ under insertion-based dispatch and are further amplified by time-dependent shortest-path evaluation. Efficiency can be improved through candidate pruning, ETA caching, and staged scoring strategies, which are standard in large-

scale dispatch systems and necessary to meet real-time decision requirements (Liang et al., 2024; Jing et al., 2025).

5. Conclusion

This study investigates the integration of ETA prediction and dispatch decision-making in on-demand delivery systems under dynamic and uncertain environments. A unified framework is proposed that combines route insertion–based dispatching, uncertainty-aware ETA estimation, and multiple decision strategies including greedy selection, Pareto-based filtering, and reinforcement learning.

Experimental results across multiple scales demonstrate that structured decision rules, particularly Pareto-based candidate filtering, consistently outperform greedy dispatching in terms of cumulative reward. This highlights the effectiveness of multi-objective reasoning in insertion-based dispatch problems, where trade-offs between delay and uncertainty play a critical role.

In contrast, the reinforcement learning approach (UETA) exhibits unstable performance, particularly in larger-scale environments, where it yields significantly lower rewards. This suggests that learning-based policies in constrained combinatorial action spaces remain sensitive to reward design, scaling, and feasibility masking. Furthermore, ablation results show that removing uncertainty-aware ETA penalties can improve performance, indicating that the current formulation of risk terms may not be well aligned with the operational objective.

Overall, the findings suggest that while integrating ETA uncertainty into dispatch decisions is conceptually promising, its practical effectiveness depends critically on proper calibration and alignment with decision objectives. This work contributes an empirical analysis of such integration and highlights key challenges in combining prediction and decision-making modules in real-time dispatch systems.

6. Discussion

The results provide several important insights into the design of integrated prediction–decision systems for real-time dispatch.

First, the consistent performance of Pareto-based selection indicates that explicit multi-objective filtering is a strong and reliable mechanism in insertion-based dispatch. By

preserving candidate solutions that balance multiple criteria, such as delay and uncertainty, Pareto methods offer a transparent and robust alternative to purely scalarized objectives. This is particularly relevant in dynamic environments where objectives may conflict and cannot be adequately captured by a single reward function.

Second, the observed instability of the reinforcement learning approach highlights the difficulty of applying policy optimization methods in combinatorial and highly constrained action spaces. Despite the use of action masking, the effective action space remains large and dynamically changing, which complicates exploration and learning. Moreover, the reward function combines operational terms with uncertainty penalties, which may introduce conflicting gradients and hinder convergence. These findings are consistent with recent literature emphasizing the sensitivity of DRL to reward design and constraint handling in real-world systems.

Third, the ablation results suggest that uncertainty-aware ETA modeling, while theoretically motivated, does not automatically translate into improved dispatch performance. The current penalty formulation based on predictive variance may overemphasize uncertainty, leading to conservative decisions that reduce overall reward. This indicates that incorporating uncertainty into decision-making requires careful calibration, and that naive integration may degrade performance rather than improve it.

Fourth, the results reveal a structural limitation of the current framework: ETA prediction and dispatch optimization are trained in a decoupled manner. While uncertainty estimates are incorporated into the decision process, the prediction model is not optimized with respect to downstream decision performance. This lack of end-to-end alignment may limit the effectiveness of the integrated system, as the ETA model is trained for predictive accuracy rather than decision utility.

From a computational perspective, the experiments also confirm that the main bottleneck lies in candidate enumeration and time-dependent ETA queries. As the system scales, the number of feasible insertion candidates grows rapidly, which increases both computational cost and decision latency. Practical deployment would therefore require additional optimization techniques such as candidate pruning, caching, or hierarchical decision strategies.

Overall, this study highlights that structured heuristics and multi-objective filtering remain strong baselines in real-time dispatch, while learning-based approaches require more careful design to achieve stable and scalable performance. Future work should explore tighter integration between prediction and decision-making, including decision-focused learning,

joint optimization of ETA and dispatch policies, and improved reward formulations that better capture operational trade-offs.

References

[1]. Berbeglia, G., Cordeau, J. F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, *202*(1), 8-15.

[2]. Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations research*, *52*(1), 35-53.

[3]. Betkier, I. (2025). Estimating travel time in transport network with a combined multi-attributed graph convolutional neural network and multilayer perceptron model. *Engineering Applications of Artificial Intelligence*, *142*, 109898.

[4]. Chen, J. F., Wang, L., Liang, Y., Yu, Y., Feng, J., Zhao, J., & Ding, X. (2024). Order dispatching via GNN-based optimization algorithm for on-demand food delivery. *IEEE Transactions on Intelligent Transportation Systems*, *25*(10), 13147-13162.

[5]. Dai, H., Gao, C., He, F., Ji, C., & Yang, Y. (2026). Optimizing driver's discount order acceptance strategies: A policy-improved deep deterministic policy gradient framework. *Transportation Research Part E: Logistics and Transportation Review*, *208*, 104628.

[6]. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

[7]. Fang, Z. Q., Sun, Q. C., Chen, L., Hu, D. L., & Gao, Y. J. (2024). Spatio-temporal learning for route-based travel time estimation. *Journal of Computer Science and Technology*, *39*(5), 1107-1122.

[8]. Gebreyesus, G., Fellek, G., Farid, A., Hou, S., Fujimura, S., & Yoshie, O. (2025). Deep reinforcement learning-based spatio-temporal graph neural network for solving job shop scheduling problem. *Evolutionary Intelligence*, *18*(1), 1-18.

[9]. Guo, B., Wang, S., Ding, Y., Wang, G., He, S., Zhang, D., & He, T. (2021, December). Concurrent order dispatch for instant delivery with time-constrained actor-critic reinforcement learning. In *2021 IEEE Real-Time Systems Symposium (RTSS)* (pp. 176-187). IEEE.

[10]. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In *International conference on machine learning* (pp. 1321-1330). PMLR.

[11]. Huang, H., Hu, C., Zhu, J., Wu, M., & Malekian, R. (2021). Stochastic task scheduling in UAV-based intelligent on-demand meal delivery system. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 13040-13054.

[12]. Huang, S., & Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. arXiv 2020. *arXiv preprint arXiv:2006.14171*.

[13]. Jin, G., Yan, H., Li, F., Huang, J., & Li, Y. (2024). Spatio-temporal dual graph neural networks for travel time estimation. *ACM Transactions on Spatial Algorithms and Systems*, *10*(3), 1-22.

[14]. Jin, Y., Song, X., Slabaugh, G., & Lucas, S. (2024). Partial advantage estimator for proximal policy optimization. *IEEE Transactions on Games*, *17*(1), 158-166.

[15]. Jing, Y., Guo, B., Li, N., Ding, Y., Liu, Y., & Yu, Z. (2025). Scalable order dispatching through federated multi-agent deep reinforcement learning. *Expert Systems with Applications*, *264*, 125792.

[16]. Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. *Advances in neural information processing systems*, *30*.

[17]. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, *30*.

[18]. Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., & Li, P. (2022, June). Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning* (pp. 11906-11917). PMLR.

[19]. Li, X., Luo, W., Yuan, M., Wang, J., Lu, J., Wang, J., ... & Zeng, J. (2021, April). Learning to optimize industry-scale dynamic pickup and delivery problems. In *2021 IEEE 37th international conference on data engineering (ICDE)* (pp. 2511-2522). IEEE.

[20]. Liang, Y., Luo, H., Duan, H., Li, D., Liao, H., Feng, J., ... & Wang, L. (2024). Meituan's real-time intelligent dispatching algorithms build the world's largest minute-level delivery network. *INFORMS Journal on Applied Analytics*, *54*(1), 84-101.

[21]. Liu, Y., Wu, F., Lyu, C., Li, S., Ye, J., & Qu, X. (2022). Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. *Transportation Research Part E: Logistics and Transportation Review*, *161*, 102694.

[22]. Mallick, T., Macfarlane, J., & Balaprakash, P. (2024). Uncertainty quantification for traffic forecasting using deep-ensemble-based spatiotemporal graph neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *25*(8), 9141-9152.

[23]. Nemirovski, A., & Shapiro, A. (2007). Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, *17*(4), 969-996.

[24]. Orda, A., & Rom, R. (1990). Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM (JACM)*, *37*(3), 607-625.

[25]. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of machine learning research*, *22*(268), 1-8.

[26]. Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.

[27]. Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, *48*, 67-113.

[28]. Sentanoe, W., Saha, S., Diyanananda, Y., Manzoor, A., Dasanayake, B., & Thyssens, D. (2024, August). Graph2RETA: Graph neural networks for pick-up and delivery route prediction and arrival time estimation. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)* (pp. 232-245). Cham: Springer Nature Switzerland.

[29]. Silva, M., Pedroso, J. P., & Viana, A. (2023). Deep reinforcement learning for stochastic last-mile delivery with crowdshipping. *EURO Journal on Transportation and Logistics*, *12*, 100105.

[30]. Staffolani, A., Darvariu, V. A., Bellavista, P., & Musolesi, M. (2025). A Cost-Aware Adaptive Bike Repositioning Agent Using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, *26*(4), 4923-4933.

[31]. Tamar, A., Chow, Y., Ghavamzadeh, M., & Mannor, S. (2015). Policy gradient for coherent risk measures. *Advances in neural information processing systems*, *28*.

[32]. Tang, L., Luo, R., Zhou, Z., & Colombo, N. (2025). Enhanced route planning with calibrated uncertainty set. *Machine Learning*, *114*(5), 129.

[33]. Wan, F., Li, L., Wang, K., Chen, L., Gao, Y., Jiang, W., & Pu, S. (2022, November). Mttpre: a multi-scale spatial-temporal model for travel time prediction. In *Proceedings of the*

30th International Conference on Advances in Geographic Information Systems (pp. 1-10).

[34]. Wang, Q., Xu, C., Zhang, W., & Li, J. (2021). GraphTTE: Travel time estimation based on attention-spatiotemporal graphs. *IEEE Signal Processing Letters, 28*, 239–243.

[35]. Wang, S., Guo, B., Ding, Y., Wang, G., He, S., Zhang, D., & He, T. (2023). Time-constrained actor-critic reinforcement learning for concurrent order dispatch in on-demand delivery. *IEEE Transactions on Mobile Computing*, *23*(8), 8175-8192.

[36]. Wang, Y., Sun, H., Lv, Y., Chang, X., & Wu, J. (2024). Reinforcement learning-based order-dispatching optimization in the ride-sourcing service. *Computers & Industrial Engineering*, *192*, 110221.

[37]. Wu, Y., Tan, H., Peter, J., Shen, B., & Ran, B. (2015). Short-term traffic flow prediction based on multilinear analysis and k-nearest neighbor regression. In *CICTP 2015* (pp. 556-569).

[38]. Zhang, L., Liu, Y., Zeng, Z., Cao, Y., Wu, X., Xu, Y., ... & Cui, L. (2024). Package Arrival Time Prediction via Knowledge Distillation Graph Neural Network. *ACM Transactions on Knowledge Discovery from Data*, *18*(5), 1-19.

[39]. Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*, 308-324.

[40]. Zhao, X., Wang, S., Wang, H., He, T., Zhang, D., & Wang, G. (2023, October). HST-GT: Heterogeneous Spatial-Temporal Graph Transformer for Delivery Time Estimation in Warehouse-Distribution Integration E-Commerce. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (pp. 3402-3411).