# Adaptive Multi-Faceted Service Capabilities Co-Prediction for Nationwide Terminal Stations in Logistics

**Shuxin Zhong[1], Kimberly Liu[2], Wenjun Lyu[3], Haotian Wang[4], Guang Wang[5], Yunhuai Liu[6], Tian He[4], Yu Yang[7], Desheng Zhang[3]**

[1] Hong Kong University of Science and Technology (Guangzhou),
[2] University of Pennsylvania,
[3] Rutgers University,
[4] JD Logistics,
[5] Florida State University,
[6] Peking University,
[7] Lehigh University

shuxinzhong@hkust-gz.edu.cn, kimliu@sas.upenn.edu, wenjun.lyu@rutgers.edu, {wanghaotian18, tim.he}@jd.com, guang@cs.fsu.edu, yunhuai.liupku.edu.cn, yuyang@lehigh.com, desheng@cs.rutgers.edu

## Abstract

Estimating service capabilities for logistics terminal stations is essential for guiding operations adjustments to enhance customer experience. However, existing studies often focus on isolated metrics like on-time delivery or complaint rates, each reflecting a specific aspect of service capabilities. To provide a more comprehensive evaluation, we design AdaService, an Adaptive multi-faceted Service capabilities co-estimation framework. We begin by constructing Multi-faceted Hypergraph to encode stations using multiple performance metrics. We then introduce a Multi-faceted Hypergraph Convolution Network (MHCN) to capture the heterogeneous service capabilities across stations, providing a comprehensive capabilities representation. Finally, we apply an Adaptive Multi-faceted Estimation module that uses multi-task learning to model dynamic interactions among these metrics, enhancing predictive accuracy. Extensive evaluation with real-world data collected from nationwide stations in a leading logistics company in China demonstrates that AdaService significantly outperforms state-of-the-art methods, improving estimation accuracy for on-time delivery, on-time pick-up, and complaint rates by up to 18.98%, 9.30%, and 39.62%.

## Introduction

In recent years, the rapid growth of e-commerce has significantly increased the demand for efficient and reliable logistics (Yu et al. 2016). As critical components of logistics networks, terminal stations are primarily responsible for parcel collection and delivery. Due to their direct interaction with customers, their service capabilities play a pivotal role in shaping customer satisfaction. To assess these capabilities, several key metrics are commonly used (Han, Chong, and Li 2020; GOV 2023): i) *On-time delivery rates*, which indicate the station's ability to handle expected workloads (Eliyan, Elomri, and Kerbache 2021). ii) *On-time pick-up rates*, which reflect the station's agility and responsiveness in meeting real-time demands (Makhloufi et al. 2015).



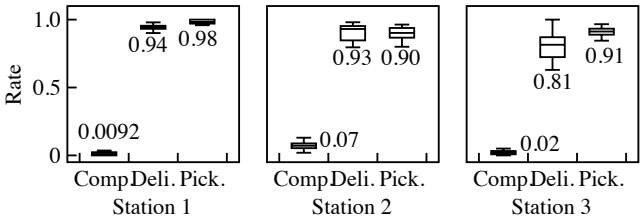


Figure 1: Heterogeneous Service Capabilities.

iii) *Complaint rates*, which provide insights into service qualities (Bruhn 2023). These metrics collectively guide logistics operations adjustments, ultimately enhancing customer experience.

However, existing research often focuses on individual metrics in isolation, without considering the interrelationships among them (Hadden et al. 2006; Yilmaz and Ari 2017; Wu and Wu 2019; Qiang et al. 2023; Zhang et al. 2023; Mao et al. 2023a). For example, a high on-time delivery rate might be achieved at the cost of increased parcel damage, which in turn leads to higher customer complaints. This narrow focus captures only a single dimension of logistics service, failing to provide a comprehensive view of overall service quality and potentially overlooking critical underlying issues (Dabholkar, Shepherd, and Thorpe 2000).

The idea sounds straightforward, two challenges persist:

- Profiling stations requires modeling diverse performance metrics, complicated by variations in geographic locations, operational processes, and customer demographics. For instance, as shown in Figure 1, Station 1 excels in both on-time delivery and pick-up rates, with low complaint rates. Station 2, despite maintaining acceptable pick-up and delivery performance but experiences higher complaint rates. Station 3 prioritizes high on-time pick-up rates, slightly compromising its delivery performance.
- The correlations among metrics are interdependent and influenced by external factors, such as delay severity and customer expectations. Figure 2 shows that in shaded re-

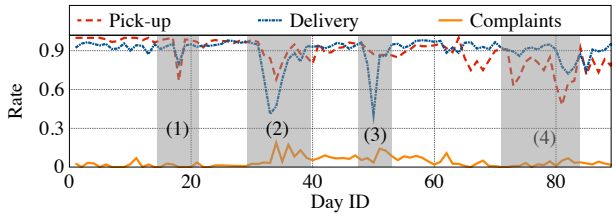


Figure 2: Uncertain Interactions Among Service Capabilities Metrics.

gion (1), declines in both on-time delivery and pick-up rates do not significantly increase complaints. Conversely, shaded regions (2) and (3) indicate that substantial decreases in either metric generally lead to more complaints. Interestingly, shaded region (4) reveals that a reduction in on-time pick-up rates does not always negatively impact on-time delivery rates.

To address these challenges, we develop `AdaService`, an Adaptive multi-faceted Service capabilities co-estimation framework for logistics terminal stations. The key intuition behind `AdaService` is that stations operating under similar conditions—such as comparable operational modes, customer bases, or market environments—are likely to exhibit similar performance. To capture station capabilities under these conditions, the hypergraph structure (Bai, Zhang, and Torr 2021) models higher-order interactions among performance metrics. Additionally, multi-task learning (MTL) (Caruana 1997) facilitates knowledge transfer across tasks, allowing for the interpretation of dynamic interactions and enhancing overall performance. In summary, we make four major contributions.

- This study is the first to co-estimate multi-faceted service capabilities for nationwide logistics terminal stations. Unlike prior work that focuses on isolated metrics, `AdaService` captures the interactions between metrics, providing a more comprehensive and context-aware view of station performance.
- We present two key technical designs. First, the Multi-faceted Hypergraph Convolution Network (MHCN) extracts features to comprehensively represent station operational abilities, capturing the complexity of station dynamics. Second, the Mixture of Experts (MoE) within an MTL framework effectively addresses metric heterogeneity, enabling precise and tailored predictions for each task.
- Extensive experiments with $770,000$ real-world records from $8,000$ stations nationwide, collected from a leading logistics company in China, demonstrate that `AdaService` outperforms state-of-the-art baselines up to $18.98\%$, $9.30\%$, and $39.62\%$ in estimating on-time delivery rate, pick-up rate, and complaint rate.

## Proposed Method

Figure 3 illustrate the overall framework, which comprises three key steps: First, we construct a Multi-faceted Hypergraph using performance embeddings learned through autoencoders (Wang, Yao, and Zhao 2016) that represent on-time delivery, pick-up, and complaint rates. Each hyperedge clusters stations based on specific performance characteristics. Next, we introduce a Multi-faceted Heterogeneous Convolution Network (MHCN) to extract features that capture the stations' operational capabilities. By iterating these processes at each time step, we model the evolving dynamics across temporal hypergraphs. Finally, a multi-task learning (MTL) approach is applied to model interactions among performance metrics, enhancing overall system performance.

### Multi-faceted Hypergraph Construction

**Operational Performance Embedding.** We use an autoencoder (Wang, Yao, and Zhao 2016) with Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997) to process performance data for different metrics, generating representations that capture distinct aspects of station capabilities. The input and output are the operational metrics for station $s_i$ over the past $t_p$ days, including on-time delivery rate $\mathbf{x}^{deli.}_{(s_i,[t-t_p:t])}$, on-time pick-up rate $\mathbf{x}^{pick.}_{(s_i,[t-t_p:t])}$ and complaints rate $\mathbf{x}^{comp.}_{(s_i,[t-t_p:t])}$.

Let us take on-time delivery rate as an example. The input sequence for the encoder is $\mathbf{x}^{deli.}_{(s_i,[t-t_p:t])} = [x^{deli.}_{(i,t-t_p)}, \ldots, x^{deli.}_{(s_i,t)}]$. Each $x^{deli.}_{(s_i,t)}$ includes day-specific information (such as holidays, weekdays, or weekends), the number of available couriers, the distribution of orders to be delivered and picked up within road areas, and the corresponding on-time delivery rate, on-time pick-up rate, and complaints rate. The intermediate embeddings $\mathbf{e}^{deli.}_{(s_i,[t-t_p:t])}$ capture the delivery capability for $s_i$ and are used by the decoder to reconstruct the input sequence. We train this autoencoder by minimizing the mean squared error (MSE) reconstruction loss. Similarly, we obtain intermediate embeddings $\mathbf{e}^{pick.}_{(s_i,[t-t_p:t])}$ and $\mathbf{e}^{comp.}_{(s_i,[t-t_p:t])}$ that represent the pick-up and service capabilities for $s_i$. All embeddings are then used in the subsequent hyperedge construction.

**Multi-faceted Hyperedge Construction.** The stations' multi-faceted capabilities embeddings at time step $t$ are denoted as $\mathbf{X}_t = [\{\mathbf{e}^{deli.}_{(s_1,t)}, \mathbf{e}^{pick.}_{(s_1,t)}, \mathbf{e}^{comp.}_{(s_1,t)}\}; \ldots; \{\mathbf{e}^{deli.}_{(s_i,t)}, \mathbf{e}^{pick.}_{(s_i,t)}, \mathbf{e}^{comp.}_{(s_i,t)}; \ldots\}]$, where $i = \{1, 2, ..., n\}$ denotes the sequence of stations. We then construct a hypergraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of stations and $\mathcal{E}$ is the set of hyperedges, with each hyperedge $e$ representing a group of stations sharing a specific characteristic. The stations in hyperedge $e_i$, denoted $Con(e_i)$, are defined as:

$$Con(e_i) = \{s_1, s_2, \ldots, s_{k_{e_i}}\} \tag{1}$$

where $k_e$ is the number of stations in $e$. Similarly, the set of hyperedges containing $s_i$, denoted as $Adj(s_i)$, is defined as:

$$Adj(s_i) = \{e_1, e_2, \ldots, e_{k_{s_i}}\} \tag{2}$$

where $k_{s_i}$ is the number of hyperedges containing $s_i$, which is considered as the centroid vertex of $Adj(s_i)$.

To construct the hyperedges, we apply the $K$-means algorithm (Jiang et al. 2019). Geographically adjacent stations are clustered using Manhattan distance (Madhulatha

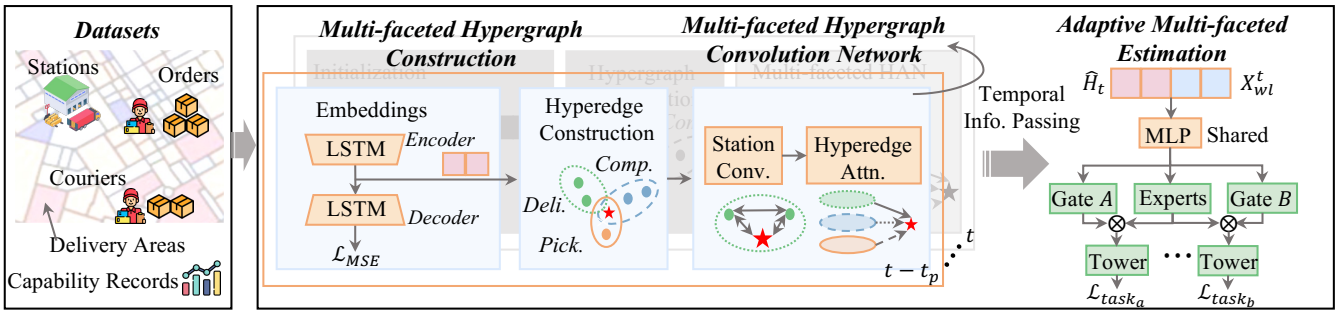


Figure 3: The Framework of AdaService. It integrates multiple data sources, including terminal stations characteristics, orders distribution, and couriers availability, to comprehensively assess logistics service capabilities. It consists of three components: (1) Multi-faceted Hypergraph Construction, (2) Multi-faceted Hypergraph Convolution Network, and (3) Adaptive Multi-faceted Estimation.

2012), while stations with similar operational characteristics are grouped using cosine distance.

## Multi-faceted Hypergraph Convolution Network

We design a Multi-faceted Hypergraph Convolution Network (MHCN) with two components-station and hyperedge convolution-for comprehensive feature extraction.

In the *station convolution* component, we apply a Graph Convolution Network (GCN) (Kipf and Welling 2016) to learn and aggregate features from individual stations. The output, termed as the hyperedge embedding, captures the intrinsic characteristics of the stations within a hyperedge. The process is mathematically defined as:

$$z_{e_i}^{(l+1)} = \sigma\Big(\sum_{s_j \in Con(e_i)} W\mathbf{e}_{s_j}^{(l)} + b\Big) \tag{3}$$

where $z_{e_i}^{(l+1)}$ is the hyperedge $e_i$ embedding, $\mathbf{e}_{s_j}^{(l)}$ is the embedding of station $s_j$ within $e_i$, $W$ is the learnable weight matrix, $b$ is the bias and $\sigma$ is the activation function.

In the *hyperedge attention* component, we use an attention mechanism (Vaswani et al. 2017) to aggregate hyperedge features into a centroid station embedding. The motivation is that the operational ability for each station is dynamically adjusted according to the varying importance of different service capabilities (Hou et al. 2019) (as illustrated in Figure 1). This step is formally represented as:

$$\begin{aligned} \alpha_{e_j}^{(l)} &= \frac{\exp(\text{LeakyReLU}(\vec{a}^\top \cdot z_{e_j}^{(l)}))}{\sum_{e_k \in \text{Adj}(s_i)} \exp(\text{LeakyReLU}(\vec{a}^\top \cdot z_{e_k}^{(l)}))}, \\ h_{s_i}^{(l+1)} &= \sum_{e_k \in \text{Adj}(s_i)} \alpha_{e_k}^{(l)} z_{e_k}^{(l)} \end{aligned} \tag{4}$$

where $\vec{a}^\top$ is a learnable weight vector, $\alpha_{e_j}^{(l)}$ is the attention score that passed through a LeakyReLU activation function to introduce non-linearity, and $h_{s_i}^{(l+1)}$ is the weighted sum of all the hyperedges $\text{Adj}(s_i)$.

The above *construction and extraction* processes are iteratively applied at each time step. To model the dynamic evolving pattern, we establish the short-term dependencies by transmitting node embeddings from adjacent time steps (Yin et al. 2022). It is formally represented by:

$$\hat{\mathbf{H}}^{t+1} = \sigma(\mathbf{H}^t \Psi + b_\psi), \tag{5}$$

$\mathbf{H}^t$ denote the stations' embeddings. $\Psi$ and $b_\psi$ are trainable parameters, and $\sigma$ is the non-linear activation function.

## Adaptive Multi-faceted Estimation

We introduce an MTL approach equipped with MoE facilitates the effective exploration of interactions between different metrics. The inputs include the estimated station capabilities $\hat{\mathbf{H}}^{t+1}$ and latest records $X_{wl}^{t+1}$, which consists of unfinished orders, available couriers, and day-specific features. The inputs are first processed by shared expert networks to extract relevant patterns.

$$f_i([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}]) = W_i \text{MLP}([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}]) \tag{6}$$

where $i = 1, 2, .., K$ and $K$ is the number of experts. Simultaneously, they are passed through the gating networks, which determine the weights assigned to each expert.

$$g_i([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}]) = \frac{exp(w_g^i \text{MLP}([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}])}{\sum_{j=1}^{K} exp(w_g^j \text{MLP}([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}])}, \tag{7}$$

where $w_g^i$ are the gating weights for the $i$-th expert. Finally, the outputs from the expert networks, weighted by the gating networks, are fed into task-specific towers, each designed to meet the unique requirements of its respective task.

$$\hat{y}_i = \sum_{j=1}^{K} g_j([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}]) f_j^i([\hat{\mathbf{H}}^{t+1}; X_{wl}^{t+1}]) \tag{8}$$

## Training Objective

In AdaService, we employ the L1 loss function as the primary training objective due to its robustness against outliers. For instance, the loss function for estimating the on-time delivery rate is defined as:

$$\mathcal{L}^{deli.}(\mathbf{W}_\varphi) = \sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{9}$$

where $\mathbf{W}_\varphi$ represents the learnable parameters, $y_i$ and $\hat{y}_i$ denote the actual and estimated on-time delivery rates, respectively. Similarly, we define loss functions for other metrics: $\mathcal{L}^{pick.}(\mathbf{W}_\varphi)$ for the on-time pick-up rate, and $\mathcal{L}^{comp.}(\mathbf{W}_\varphi)$ for the complaints rate.

The overall loss function combines these losses:

$$\mathcal{L}^{total} = w_{deli.}\mathcal{L}^{deli.} + w_{pick.}\mathcal{L}^{pick.} + w_{comp.}\mathcal{L}^{comp.}, \quad (10)$$

where $w_{deli.}$, $w_{pick.}$, and $w_{comp.}$ are weights that adapt to the importance of each task in stations' service capabilities.

# Experiments

This section presents the data and evaluation settings, followed by a detailed analysis of the evaluation results.

## Experimental Setup

**Datasets** Our evaluation utilizes a dataset from one of the major logistics companies in China, collected from July 1 to August 1, 2023. This dataset includes 770,000 performance records from 8,000 stations. To comprehensively analyze station service capabilities, we focus on extracting both dynamic and static features.

- *Static features* include GPS coordinates and delivery area characteristics for each station. GPS coordinates are derived from a geo-coding system that decodes textual addresses from waybills. Delivery areas are segmented based on attributes such as size, classification (e.g., residential, industrial), delivery difficulty, and distance from the station.
- *Dynamic features* consist of time-varying operational data, including the daily number of available couriers, the daily volume of pick-up and delivery orders within delivery areas, and the station's daily performance metrics (i.e., on-time delivery, on-time pick-up, and complaint rates).

During Data Pre-processing, we applied Min-Max normalization to scale numerical data (e.g., order volume) to a $[0, 1]$ range, and used one-hot encoding for categorical data (e.g., station types). After estimation, the values were reverted to their original scales for evaluation.

In the initial operational performance learning phase, we used performance records from the preceding 5 days to learn stations' capabilities. The hyperparameters for `AdaService` were set as follows: embedding hidden size at 256, graph convolution filter size at 1, and historical data length at 2. `AdaService` was optimized using the Adam optimizer (Kingma and Ba 2014), with a batch size of 128 and a learning rate of 0.001.

For evaluation, we employed standard metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). To align with the company's practical needs—specifically, whether the predicted values fall within an acceptable error range—we defined specific accuracy metrics: Delivery Accuracy (Deli. Acc.), Pick-up Accuracy (Pick-up Acc.), and Complaints Accuracy (Comp. Acc.) as:

$$Acc(i) = \begin{cases} 1, & |y_i - \hat{y}_i| < \epsilon \\ 0, & \text{otherwise}, \end{cases} \quad (11)$$

where $y_i$ represents the actual value, $\hat{y}_i$ is the predicted value, and $\epsilon$ denotes an acceptable deviation threshold.

**Baselines** To evaluate the performance of `AdaService`, we compare it against a range of baselines, from traditional statistical methods to contemporary neural network-based models. The comparative models include:

- **Historical Average (HA)**: It calculates average on-time delivery and pick-up, and complaint rates to estimate the future service capabilities, which does not account for time-varying factors such as daily workload.
- **XGBoost (Chen and Guestrin 2016)**: Uses geographical and workload features of stations as inputs to predict service capabilities. XGBoost excels at identifying performance-influencing factors.
- **MLP (Bishop 1995)**: Similar to XGBoost in terms of inputs and outputs, but specializes in capturing complex, non-linear patterns.
- **LSTM (Hochreiter and Schmidhuber 1997)**: It processes sequential data to estimate the evolving patterns of station service capabilities.
- **STGCN (Yan, Xiong, and Lin 2018)**: The stations are represented using nodes and the adjacent matrix is calculated based on proximity, which benefits to capture similar service capabilities from neighboring stations.
- **DyHCN (Yin et al. 2022)**: A dynamic hypergraph-based model that combines hypergraph construction, spatiotemporal hypergraph convolution, and a collaborative estimation module to facilitate evolving process estimation.
- **Multi-task Learning (MTL) (Caruana 1997)**: Similar to XGBoost in terms of inputs but outputs are multiple service metrics. MTL leverages shared layers for common feature learning and task-specific layers for individual task nuances, enhancing overall performance.

## Overall Performance

We compared our proposed `AdaService` model with baseline models, as summarized in Table 1. For a fairness comparison, we implement `AdaService-`, a variant of our model without the MTL module. Notably, `AdaService` outperforms all baselines in key metrics.

The *Historical Average (HA)* method estimates next-day service capabilities based on average historical performance of corresponding metrics (e.g., on-time delivery rate), but it overlooks dynamic factors like courier availability and daily workload. In constrast, `Xgboost`(Chen and Guestrin 2016) and `MLP`(Popescu et al. 2009) address this by incorporating daily order volumes and courier availability, yet they are outperformed by `LSTM`, which underscores the importance of temporal correlations. Moreover, `STGCN` improves upon these by integrating geographical correlations, while `DyHCN` and `AdaService-` achieve even better results by incorporating semantic information. Finally, `AdaService-` slightly outperforms `DyHCN` due to its more comprehensive representation of station capabilities.

Additionally, we enhanced the models with MTL (i.e., `LSTM+MTL`, `STGCN+MTL`, and `DyHCN+MTL`), resulting in improved performance. It highlights the effectiveness of

| Models | Metrics | | | | | |
|---|---|---|---|---|---|---|
| | **MSE** | **MAE** | **MAPE (%)** | **Delivery Accuracy** | **Pick-up Accuracy** | **Complaints Accuracy** |
| XGBoost (Chen and Guestrin 2016) | 0.92 | 0.031 | 0.67 | 0.79 | 0.43 | 0.53 |
| MLP (Bishop 1995) | 0.86±0.022 | 0.026±0.002 | 0.62±0.2 | 0.67±0.012 | 0.41±0.015 | 0.49±0.018 |
| LSTM (Hochreiter and Schmidhuber 1997) | 0.81±0.017 | 0.022±0.001 | 0.57±0.3 | 0.86±0.011 | 0.36±0.012 | 0.65±0.010 |
| STGCN (Yan, Xiong, and Lin 2018) | 0.76±0.015 | 0.015±0.002 | 0.48±0.2 | 0.90±0.015 | 0.43±0.011 | 0.69±0.015 |
| DyHCN (Yin et al. 2022) | 0.77±0.018 | 0.016±0.013 | 0.46±0.2 | 0.91±0.012 | 0.44±0.011 | 0.71±0.013 |
| AdaService- | 0.75±0.015 | 0.015±0.011 | 0.45±0.2 | 0.93±0.008 | 0.46±0.011 | 0.72±0.016 |
| LSTM+MTL | 0.78±0.014 | 0.018±0.005 | 0.49±0.2 | 0.89±0.008 | 0.30±0.015 | 0.67±0.012 |
| STGCN+MTL | 0.74±0.013 | 0.014±0.005 | 0.45±0.1 | 0.91±0.007 | 0.46±0.012 | 0.71±0.009 |
| DyHCN+MTL | 0.74±0.015 | 0.014±0.009 | 0.44±0.2 | 0.92±0.013 | 0.46±0.017 | 0.73±0.014 |
| AdaService | **0.70±0.017** | **0.010±0.003** | **0.39±0.2** | **0.96±0.012** | **0.49±0.015** | **0.77±0.015** |

Table 1: Performance Comparison of AdaService with Baseline Methods. w.r.t RMSE, MAE, MAPE, Accuracies for On-time Delivery Rate, On-time Pick-up Rate, and Complaint Rate with Standard Deviations. Deli. Acc., Pick. Acc., and Comp. Acc. represents delivery rate, pick-up rate, and complaints number accuracy, respectively. The **best** results are highlighted.

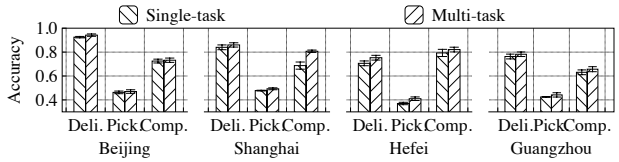


Figure 4: The Impact of City.

MTL in modeling the dynamic interactions and dependencies among performance of various metrics.

## Impact of City Scale on AdaService

Furthermore, we evaluated AdaService in several representative cities [1], including Beijing (the capital of China), Shanghai (a metropolitan city of China), Chengdu and Hefei (two mid-sized cities in China), as shown in Figure 4. We notice, in major cities like Beijing and Shanghai, we observed a modest improvement in delivery accuracy of approximately 2.7% and 3.7%. In contrast, mid-sized cities like Hefei showed a more significant increase of 7.1%. The limited improvement in larger cities is likely due to their stable and high performance in on-time delivery and pick-up rates, supported by a sufficient number of couriers. On the other hand, mid-sized cities often struggle to balance high order volumes with limited courier capacities, leading to trade-offs between on-time delivery and pick-up rates. As a result, the performance improvement from collectively analyzing these metrics is more pronounced in mid-sized cities.

## Ablation Studies

Moreover, we evaluated the peformance of key components, i.e., MHCN and MTL, within AdaService.

[1]https://en.wikipedia.org/wiki/List_of_Chinese_prefecture-level_cities_by_GDP

| Metrics | Models | | | |
|---|---|---|---|---|
| | **∧MLP** | **∧LSTM** | **∧STGCN** | **AdaService** |
| MSE | 0.86±0.022 | 0.81±0.015 | 0.76±0.017 | **0.70±0.017** |
| MAE | 0.026±0.002 | 0.022±0.001 | 0.015±0.002 | **0.010±0.003** |
| MAPE (%) | 0.62±0.2 | 0.57±0.3 | 0.48±0.2 | **0.39±0.2** |
| Deli. Acc. | 0.81±0.021 | 0.86±0.016 | 0.90±0.015 | **0.96±0.012** |
| Pick. Acc. | 0.31±0.015 | 0.36±0.015 | 0.43±0.013 | **0.49±0.012** |
| Comp. Acc. | 0.62±0.021 | 0.65±0.018 | 0.69±0.015 | **0.77±0.015** |

Table 2: Performance for Different Encoding Strategies.

First, we replaced the MHCN with three variants:

- ∧MLP, which uses the current state of a single station;
- ∧LSTM, which captures a station's time-varying state;
- ∧STGCN, which considers neighboring stations' states.

The embeddings from these variants were then fed into the designed MTL framework used in AdaService. The results, shown in Table 2, indicate that ∧LSTM, which captures the temporal trends, outperforms ∧MLP, which merely considers the current state. ∧STGCN further improves performance, incorporating the similar patterns from neighboring stations. However, AdaService outperforms ∧STGCN, highlighting the critical role of semantic correlations in learning station capabilities representation.

Second, using the outputs from MHCN, we evaluated two MTL variants: hard parameter sharing (Hard) and Task Relation-based MTL (Task). The results are shown in Table 3. Task outperforms Hard by better capturing the significant differences between tasks. Furthermore, AdaService surpasses Task due to its ability to model dynamic and uncertain interactions between tasks.

| Metrics | Models | | |
|---|---|---|---|
| | `Hard` | `Task` | `AdaService` |
| MSE | 0.88±0.018 | 0.82±0.015 | **0.70±0.017** |
| MAE | 0.031±0.001 | 0.023±0.001 | **0.010±0.003** |
| MAPE (%) | 0.64±0.3 | 0.58±0.2 | **0.39±0.2** |
| Deli. Acc. | 0.79±0.016 | 0.87±0.015 | **0.96±0.015** |
| Pick. Acc. | 0.29±0.018 | 0.36±0.013 | **0.49±0.012** |
| Comp. Acc. | 0.61±0.018 | 0.64±0.016 | **0.77±0.015** |

Table 3: Performance for Different Aggregation Strategies.

| Metrics | Length of Historical Data ($l$) | | | |
|---|---|---|---|---|
| | $l = 2$ | $l = 3$ | $l = 4$ | $l = 5$ |
| MSE | **0.70±0.017** | 0.76±0.011 | 0.79±0.015 | 0.76±0.016 |
| MAE | **0.010±0.003** | 0.016±0.001 | 0.021±0.002 | 0.023±0.002 |
| MAPE (%) | **0.39±0.2** | 0.49±0.18 | 0.52±0.20 | 0.54±0.15 |
| Deli. Acc. | **0.92±0.020** | 0.89±0.018 | 0.86±0.015 | 0.82±0.014 |
| Pick. Acc. | **0.47±0.022** | 0.44±0.017 | 0.41±0.012 | 0.38±0.016 |
| Comp. Acc. | **0.77±0.015** | 0.69±0.018 | 0.65±0.015 | 0.62±0.018 |

Table 4: Performance for Different Historical Steps.

## Parameters Sensitivities

We explore how various parameters influence the performance of `AdaService`, as shown in Table 4. Typically, we focus on the following parameters, each analyzed individually while keeping others at their default settings:

- Length of historical data ($l$) for capabilities learning.
- Number of hypergraph convolution layers ($n_l$).
- Geographic distance threshold ($d$) for station correlation.
- Number of clusters initialized for K-means ($k_c$).

Based on our analysis, we have the following observations: i) `AdaService` is generally sensitive to parameter adjustments. Significant performance variations occur with different lengths of historical data and numbers of hypergraph convolution layers. The optimal length of historical data is 2 days; longer periods may introduce irrelevant data, detracting from performance (Table 4). Similarly, a single hypergraph convolution layer yields the best results, as additional layers can overcomplicate the model (Table 5). ii) The geographic distance threshold is crucial for station correlation. A low threshold (e.g., $5,000$ m) limits relevant data, while a high threshold (beyond $10,000$ m) introduces noise, both leading to suboptimal performance (Table 6). iii) The number of clusters initialized for K-means, ranging from $15$ to $60$, shows that `AdaService`'s performance is relatively stable across this parameter, indicating the robustness of the hypergraph construction (Table 7).

| Metrics | Number of Hypergraph Convolutional Layers ($n_l$) | | | |
|---|---|---|---|---|
| | $n_l = 1$ | $n_l = 3$ | $n_l = 5$ | $n_l = 7$ |
| MSE | **0.70±0.017** | 0.75±0.020 | 0.79±0.015 | 0.82±0.012 |
| MAE | **0.010±0.003** | 0.015±0.002 | 0.017±0.001 | 0.020±0.002 |
| MAPE (%) | **0.39±0.2** | 0.51±0.16 | 0.57±0.19 | 0.61±0.15 |
| Deli. Acc. | **0.96±0.012** | 0.90±0.011 | 0.87±0.015 | 0.85±0.012 |
| Pick. Acc. | **0.49±0.015** | 0.43±0.012 | 0.41±0.015 | 0.40±0.012 |
| Comp. Acc. | **0.77±0.015** | 0.70±0.016 | 0.69±0.015 | 0.67±0.018 |

Table 5: Performance for Different Convolution Layers.

| Metrics | Geographical Distance Thresholds ($d$) | | | |
|---|---|---|---|---|
| | $d = 5,000$ | $d = 10,000$ | $d = 15,000$ | $d = 20,000$ |
| MSE | 0.76±0.018 | **0.70±0.017** | 0.79±0.015 | 0.82±0.018 |
| MAE | 0.018±0.002 | **0.010±0.003** | 0.017±0.001 | 0.021±0.002 |
| MAPE (%) | 0.53±0.11 | **0.39±0.2** | 0.49±0.18 | 0.56±0.17 |
| Deli. Acc. | 0.81±0.017 | **0.96±0.012** | 0.90±0.013 | 0.88±0.015 |
| Pick. Acc. | 0.30±0.012 | **0.49±0.015** | 0.42±0.015 | 0.38±0.019 |
| Comp. Acc. | 0.61±0.022 | **0.77±0.015** | 0.70±0.015 | 0.63±0.021 |

Table 6: Performance on Different Distance Thresholds.

| Metrics | Number of Clusters ($k_c$) | | | |
|---|---|---|---|---|
| | $k_c = 15$ | $k_c = 30$ | $k_c = 45$ | $k_c = 60$ |
| MSE | 0.771±0.016 | **0.70±0.017** | 0.742±0.015 | 0.784±0.015 |
| MAE | 0.018±0.002 | **0.010±0.003** | 0.016±0.001 | 0.019±0.002 |
| MAPE (%) | 0.51±0.19 | **0.39±0.2** | 0.48±0.17 | 0.53±0.15 |
| Deli. Acc. | 0.83±0.015 | **0.96±0.012** | 0.87±0.015 | 0.84±0.016 |
| Pick. Acc. | 0.35±0.012 | **0.49±0.015** | 0.38±0.015 | 0.36±0.019 |
| Comp. Acc. | 0.64±0.017 | **0.77±0.015** | 0.68±0.015 | 0.65±0.016 |

Table 7: Performance for Different Numbers of Clusters.

# Discussion

In this section, we first discuss two lessons learned from this work, followed by limitations and future works. Finally, we show the generalization and privacy issues analysis.

## Insights and Lessons Learned

Based on the design and real-world experiments of `AdaService`, we summarize the following lessons:

- **Effectiveness of Multi-task Co-assessment.** `AdaService` integrates multi-task learning to uncover complex interconnections within multi-faceted metrics. As shown in Figure 3, the implementation of this design results in performance enhancement of $16.2\%$. This approach leads to a performance enhancement of $16.2\%$, verifying that learning a compact and generalized

representation of a task is achievable by recognizing and extracting commonalities across different tasks.

- **Impact of City Scale on Performance:** Our study revealed significant performance variations of `AdaService` across different urban scales (supported by Figure 4). For instance, `AdaService` showed a 7.1% improvement in Hefei, a mid-sized city, compared to a 2.7% improvement in Beijing, a larger metropolis. This difference is partly due to the strong correlations between delivery and pick-up efficiency in mid-sized cities, where limited courier numbers and larger service areas present challenges. Thus, a comprehensive analysis incorporating various metrics, such as on-time delivery and pick-up rates, is crucial for enhancing performance, particularly in smaller cities.

### Generalization and Implication

We explore multi-faceted operational metrics to evaluate the service capabilities of logistics terminal stations, a method that can be generalized to other scenarios. For example, in a taxi dispatch system, we can simultaneously estimate passenger alighting and boarding volumes, idle taxi rates, and response times to assess the service capabilities of taxis in different areas. It helps efficiently redistribute idle taxis from high-capacities areas to low-capacities areas.

### Ethics and Privacy

To protect data privacy, we first convert each order's address into GPS coordinates, effectively masking detailed customer information. Secondly, we remove unnecessary data fields to minimize exposure. Finally, data access is restricted to authorized core team members, ensuring secure handling.

## Related Work

### Service Capability Estimation

Current research focuses on estimating service capabilities from a single perspective (Wu and Wu 2019; Wen et al. 2022b; Hadden et al. 2006; Yilmaz and Ari 2017). Specifically, in logistics, studies (Wen et al. 2022a; Mao et al. 2023b,a) utilize on-time delivery or pick-up rates as key indicators. DeepETA (Wu and Wu 2019) enhances delivery time prediction by integrating historical data. Similarly, Graph2Route (Wen et al. 2022a) and DeepRoute+(Wen et al. 2022b) focus on forecasting delivery routes and times, incorporating couriers' behaviors (Hong et al. 2024) under time constraints. GMDNet(Mao et al. 2023a) and DRL4Route (Mao et al. 2023b) are geared towards optimizing delivery and pick-up times to boost last-mile logistics efficiency. In other fields, Hadden (Hadden et al. 2006) examines customer churn as a reflection of service quality, while (Yilmaz and Ari 2017) assesses rail service quality through customer feedback.

### Hypergraph

We employ Hypergraph for non-pairwise correlations between stations (Ma et al. 2022; Sun et al. 2021). For instance,(Sun et al. 2021) and(Ma et al. 2022) utilize hypergraphs to reveal complex relationships within social networks and for causal analysis in epidemiology. Similarly, (Zhao et al. 2023) uses hyperedges to examine the impact of external events and neighborhood dynamics over time. DyHCN (Yin et al. 2022) implements a dynamic Hypergraph Convolutional Network with an attention mechanism for time series prediction. Furthermore, (Saifuddin et al. 2023) employs a hypergraph constructed from drug chemical substructures to explore drug similarities.

### Multi-task Learning

We utilize MTL to reveal complex interconnections between interrelated tasks (Caruana 1997; Sun et al. 2021; Wang et al. 2022). For instance, (Cai et al. 2023) introduced a multi-layered graph model for route and time co-prediction in logistics. (Wang et al. 2022) developed a model with diverse experts to investigate task relationships and identify task-specific features. (Zhai et al. 2023) utilized MTL to analyze correlations in group buying recommendations, breaking it down into two connected sub-tasks. Additionally, (Zhou et al. 2023) created a hierarchical information extraction network to effectively manage the complex interplay of relationships across various scenarios and tasks.

## Conclusion

In this paper, we present `AdaService`, an adaptive, multi-faceted service capabilities co-estimation framework for logistics terminal stations, designed to simultaneously estimate key operational metrics: on-time delivery, on-time pick-up, and complaint rates. Specifically, `AdaService` comprises two technical designs: the *Multi-faceted Hypergraph Convolution Network*, which generates a comprehensive representation of each station's service capabilities, and the *Adaptive Multi-faceted Estimation*, which captures dynamic interactions among these metrics. Our evaluation using real-world data from one of the biggest logistics companies in China demonstrates that `AdaService` significantly improves estimation performance for on-time delivery, on-time pick-up, and complaint rates by 18.98%, 9.30%, and 39.62%, respectively. To further validate `AdaService`'s effectiveness, we use its outputs to guide the cross-station courier re-scheduling simulation resulted in a 2.11% improvement in on-time delivery rates, while maintaining stable on-time pick-up and complaints rates.

## Acknowledgments

This work is supported partly by National Key Research Plan under grant No.2021YFB2900100, the National Natural Science Foundation of China (NSFC) 61925202, the Jiangsu Provincial Key Research and Development Program under Grant BE2022065-1, BE2022065-3, Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things(No.2023B1212010007), China NSFC Grant(U2001207, 62472366), and the Project of DEGP (No.2024GCZX003, 2023KCXTD042, 2021ZDZX1068).

## References


Bai, S.; Zhang, F.; and Torr, P. H. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637.

Bishop, C. M. 1995. *Neural networks for pattern recognition*. Oxford university press.

Bruhn, M. 2023. Measuring service quality. In *Quality Management for Services: Handbook for Successful Quality Management. Principles–Concepts–Methods*, 141–226. Springer.

Cai, T.; Wan, H.; Wu, F.; Wen, H.; Guo, S.; Wu, L.; Hu, H.; and Lin, Y. 2023. M 2 G4RTP: A Multi-Level and Multi-Task Graph Model for Instant-Logistics Route and Time Joint Prediction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 3296–3308. IEEE.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28: 41–75.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Dabholkar, P. A.; Shepherd, C. D.; and Thorpe, D. I. 2000. A comprehensive framework for service quality: an investigation of critical conceptual and measurement issues through a longitudinal study. *Journal of retailing*, 76(2): 139–173.

Eliyan, A.; Elomri, A.; and Kerbache, L. 2021. The last-mile delivery challenge: Evaluating the efficiency of smart parcel stations. In *Supply Chain Forum: An International Journal*, volume 22, 360–369. Taylor & Francis.

GOV. 2023. Complaints. https://www.spb.gov.cn/gjyzj/c100015/c100016/202311/06eeb618a6814689ad947244817314bb.shtml. Accessed: 2023-11-17.

Hadden, J.; Tiwari, A.; Roy, R.; and Ruta, D. 2006. Churn prediction using complaints data. In *Proceedings of world academy of science, engineering and technology*, volume 19, 158–163.

Han, Y.; Chong, W. K.; and Li, D. 2020. A systematic literature review of the capabilities and performance metrics of supply chain resilience. *International Journal of Production Research*, 58(15): 4541–4566.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hong, Z.; Li, Z.; Zhong, S.; Lyu, W.; Wang, H.; Ding, Y.; He, T.; and Zhang, D. 2024. CrossHAR: Generalizing Cross-dataset Human Activity Recognition via Hierarchical Self-Supervised Pretraining. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(2).

Hou, R.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2019. Cross attention network for few-shot classification. *Advances in neural information processing systems*, 32.

Jiang, J.; Wei, Y.; Feng, Y.; Cao, J.; and Gao, Y. 2019. Dynamic Hypergraph Neural Networks. In *IJCAI*, 2635–2641.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Ma, J.; Wan, M.; Yang, L.; Li, J.; Hecht, B.; and Teevan, J. 2022. Learning causal effects on hypergraphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1202–1212.

Madhulatha, T. S. 2012. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

Makhloufi, R.; Cattaruzza, D.; Meunier, F.; Absi, N.; and Feillet, D. 2015. Simulation of mutualized urban logistics systems with real-time management. *Transportation Research Procedia*, 6: 365–376.

Mao, X.; Wan, H.; Wen, H.; Wu, F.; Zheng, J.; Qiang, Y.; Guo, S.; Wu, L.; Hu, H.; and Lin, Y. 2023a. GMDNet: A Graph-Based Mixture Density Network for Estimating Packages' Multimodal Travel Time Distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4561–4568.

Mao, X.; Wen, H.; Zhang, H.; Wan, H.; Wu, L.; Zheng, J.; Hu, H.; and Lin, Y. 2023b. DRL4Route: A Deep Reinforcement Learning Framework for Pick-up and Delivery Route Prediction. *arXiv preprint arXiv:2307.16246*.

Popescu, M.-C.; Balas, V. E.; Perescu-Popescu, L.; and Mastorakis, N. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7): 579–588.

Qiang, Y.; Wen, H.; Wu, L.; Mao, X.; Wu, F.; Wan, H.; and Hu, H. 2023. Modeling Intra-and Inter-community Information for Route and Time Prediction in Last-mile Delivery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 3106–3112. IEEE.

Saifuddin, K. M.; Bumgardner, B.; Tanvir, F.; and Akbas, E. 2023. Hygnn: Drug-drug interaction prediction via hypergraph neural network. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 1503–1516. IEEE.

Sun, X.; Yin, H.; Liu, B.; Chen, H.; Meng, Q.; Han, W.; and Cao, J. 2021. Multi-level hyperedge distillation for social linking prediction on sparsely observed networks. In *Proceedings of the Web Conference 2021*, 2934–2945.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, S.; Li, Y.; Li, H.; Zhu, T.; Li, Z.; and Ou, W. 2022. Multi-task learning with calibrated mixture of insightful experts. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3307–3319. IEEE.

Wang, Y.; Yao, H.; and Zhao, S. 2016. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184: 232–242.

Wen, H.; Lin, Y.; Mao, X.; Wu, F.; Zhao, Y.; Wang, H.; Zheng, J.; Wu, L.; Hu, H.; and Wan, H. 2022a. Graph2Route: A Dynamic Spatial-Temporal Graph Neural Network for Pick-up and Delivery Route Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4143–4152.

Wen, H.; Lin, Y.; Wan, H.; Guo, S.; Wu, F.; Wu, L.; Song, C.; and Xu, Y. 2022b. DeepRoute+: Modeling Couriers' Spatial-temporal Behaviors and Decision Preferences for Package Pick-up Route Prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2): 1–23.

Wu, F.; and Wu, L. 2019. DeepETA: a spatial-temporal sequential neural network model for estimating time of arrival in package delivery system. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 774–781.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Yilmaz, V.; and Ari, E. 2017. The effects of service quality, image, and customer satisfaction on customer complaints and loyalty in high-speed rail service in Turkey: a proposal of the structural equation model. *Transportmetrica A: Transport Science*, 13(1): 67–90.

Yin, N.; Feng, F.; Luo, Z.; Zhang, X.; Wang, W.; Luo, X.; Chen, C.; and Hua, X.-S. 2022. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 1621–1634. IEEE.

Yu, Y.; Wang, X.; Zhong, R. Y.; and Huang, G. Q. 2016. E-commerce logistics in supply chain management: Practice perspective. *Procedia Cirp*, 52: 179–185.

Zhai, S.; Liu, B.; Yang, D.; and Xiao, Y. 2023. Group Buying Recommendation Model Based on Multi-task Learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 978–991. IEEE.

Zhang, L.; Zhou, X.; Zeng, Z.; Cao, Y.; Xu, Y.; Wang, M.; Wu, X.; Liu, Y.; Cui, L.; and Shen, Z. 2023. Delivery Time Prediction Using Large-Scale Graph Structure Learning Based on Quantile Regression. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 3403–3416. IEEE.

Zhao, Y.; Luo, X.; Ju, W.; Chen, C.; Hua, X.-S.; and Zhang, M. 2023. Dynamic Hypergraph Structure Learning for Traffic Flow Forecasting. ICDE.

Zhou, J.; Cao, X.; Li, W.; Bo, L.; Zhang, K.; Luo, C.; and Yu, Q. 2023. HiNet: Novel Multi-Scenario & Multi-Task Learning with Hierarchical Information Extraction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 2969–2975. IEEE.