LMGL-WD: LLM-Guided Multi-Task Graph Learning for Category-Level Warehouse Demand Prediction in E-Commerce

Wenjun Lyu[1,2,*,†], Fangyu Li[1,3,*], Yudong Zhang[4], Shuai Wang[5], Yunhuai Liu[6], Tian He[1], Desheng Zhang[2]

[1] JD Logistics
[2] Rutgers University
[3] Huazhong University of Science and Technology
[4] University of Science and Technology of China
[5] Southeast University
[6] Peking University

Abstract

In warehouse-based e-commerce, accurate category-level warehouse demand prediction is essential to ensure effective inventory management. Existing works mainly explore advanced time series models to capture the temporal dynamics, failing to mine cross-category and cross-warehouse correlations effectively. In this paper, we explore large language models to understand the semantic information and fuse multi-view knowledge to enhance demand prediction. However, it is not trivial due to: i) the inaccurate LLM's understanding of the category-related and warehouse-related textual input; and ii) the complicated cross-warehouse knowledge utilization. To solve the above challenges, we propose an LLM-guided multi-task graph learning framework, LMGL-WD, for category-level warehouse demand prediction. Specifically, LMGL-WD includes three components: i) an LLM-guided category series encoding module to represent each category through contextual and series embedding; ii) a cross-warehouse category learning module to adaptively mine the informative knowledge from cross-warehouses to enhance category representation; and iii) a cross-category multi-task learning module to adaptively capture cross-category correlations to improve demand prediction. Evaluation results with real-world data from one of the largest e-commerce platforms in China demonstrate that LMGL-WD achieves superior performance, e.g., reduces MAPE by up to 31.59%, compared to state-of-the-art methods.

Introduction

Warehouse-based e-commerce platforms, such as Amazon in the U.S. (Amazon 2025) and JD Retail in China (JD-Retail 2025), have replaced the traditional single-origin shipping model with distributed networks of warehouses, experiencing significant development in recent years. Each customer order is generally shipped from the closest or fastest-to-serve warehouse that holds the required product inventory for the specified category, thereby minimizing the order delivery time. The key operational challenge in warehouse-based e-commerce is to keep every warehouse's appropriate inventory for each category: excess stock raises holding costs and compresses margins, while insufficient stock increases delivery times and decreases customer satisfaction. To align inventory with the distinct demand pattern of each category and avoid costly surpluses or service-damaging shortages, it is essential to achieve accurate warehouse demand prediction at the category level in warehouse-based e-commerce.

Most existing research works on category demand prediction have mainly concentrated on refining deep-learning-based time series models to capture the temporal dynamics (Bandara et al. 2019; Qi et al. 2019; Singh et al. 2020; Shi et al. 2021). However, these studies fail to exploit the intricate cross-category and cross-warehouse correlations effectively, leading to the sub-optimal prediction performance at the warehouse-category granularity.

In this work, we explore the large language model (LLM) to capture the informative category representation through understanding the semantic information about categories and warehouse descriptions. The LLM-enhanced category representation can then be effectively utilized for cross-warehouse, cross-category information fusion to improve the model's prediction capability.

However, there exist two main challenges to perform LLM-enhanced category-level warehouse demand prediction: i) *Inaccurate LLM Understanding*: LLM-based category and warehouse textual information understanding may provide an inaccurate category representation. An effective method needs to be designed to explore the LLM's advantages in dealing with contextual inputs; and ii) *Complicated Cross-Warehouse Information Utilization*: Although cross-warehouse features can be complementary information to learn the category demand, indiscriminate information aggregation risks injecting noise. It is not trivial to judiciously select the knowledge from appropriate warehouses to strengthen prediction performance.

To solve the above challenges, we propose an LLM-guided multi-task graph learning framework, called

*These authors contributed equally.
†Corresponding author.

LMGL−WD, for category-level warehouse demand prediction in warehouse-based e-commerce. Specifically, we design a category-level contextual encoder and a category-level series encoder to utilize the series data and LLM understanding capacity effectively for the category representation. We also propose an adaptive similarity measurement scheme based on the Origin-Destination (OD)-level demand distributions and geographic information of warehouses to capture the most informative cross-warehouse knowledge effectively. The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to explore the category-level warehouse demand prediction in warehouse-based e-commerce with the assistance of LLM, which is significant to avoid inventory surpluses or service-damaging shortages of various categories.
- We design a demand prediction framework, LMGL−WD, for warehouse-based e-commerce, which includes three main components: i) an LLM-guided category series encoding module to obtain the effective representation of various categories; ii) a cross-warehouse category learning module to adaptively mine informative knowledge from cross-warehouses to complement the category representation; and iii) a cross-category multi-task learning module to adaptively fuse cross-category correlations for cooperative demand prediction.
- We implement our framework LMGL−WD with real-world warehouse-related and order-related data collected from one of the largest e-commerce platforms in China. Extensive evaluation results demonstrate that LMGL−WD outperforms state-of-the-art methods, e.g., improving MAPE by up to 31.59%.

Related Work

Time Series Prediction

Most research on time series prediction captures correlations in time series data using various neural networks (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Liu et al. 2022; Challu et al. 2023; Nie et al. 2022). However, these methods fail to effectively integrate additional contextual information into time series, resulting in suboptimal prediction performance. In recent years, large language models (LLMs) have demonstrated strong cross-domain semantic understanding capabilities, and some studies have explored the application of LLMs in time series prediction (Jin et al. 2024; Cao et al. 2023; Sun et al. 2023; Pan et al. 2024). For example, Jin et al. (Jin et al. 2023) propose a reprogramming framework that aligns the time series data and natural language representation to improve time series prediction performance. Liu et al. (Liu et al. 2024) design a cross-modality alignment method to utilize time series encoding and prompt embedding effectively for multivariate time series prediction.

Graph Neural Network

Graph Neural Network (GNN) has become a hot research topic in the past several years (Huang et al. 2025a; Fu et al. 2025; Wu et al. 2020), and has been applied in various areas. For example, Wang et al. (Wang et al. 2024) and Cai et al. (Cai et al. 2023b) explore GNN for recommendation systems. Zhong et al. (Zhong et al. 2025) use a hypergraph-based framework to predict the multi-faceted capabilities of logistics terminal stations. Yang et al. (Yang et al. 2024) design a hypergraph convolutional network to capture multi-source contextual information for illegal parking prediction in cities. Kong et al. (Kong et al. 2024) propose a graph-based network for traffic prediction. Feng et al. (Feng et al. 2025) design a heterogeneous GNN for customer expansion tasks in the urban logistics industry. Zhang et al. (Zhang et al. 2023a) utilize a graph learning method to predict the estimated time of arrival for packages in e-commerce. In this work, we explore GNN to capture the informative knowledge across warehouses.

Multi-Task Learning

Multi-task learning enhances the model's performance on each task by simultaneously learning multiple tasks and effectively leveraging the knowledge provided by other tasks. For example, Cai et al. (Cai et al. 2023a) propose a multi-layer graph model for joint prediction of a courier's route and times to various locations in urban logistics scenarios. Wang et al. (Wang et al. 2022) develop a model containing multiple experts to study inter-task relationships and identify task-specific features. Zhai et al. (Zhai et al. 2023) use multi-task learning to analyze correlations in group buying recommendations, decomposing it into two interrelated sub-tasks. Hu et al. (Hu et al. 2023) find high-risk locations in epidemics with public health data and multi-task learning. Guo et al. (Guo et al. 2024) explore multi-task learning to enhance the logistics advertising conversion prediction. Huang et al. (Huang et al. 2025b) explore multi-task fine-tuning methods to enhance the generalization ability of LLMs in multi-task scenarios. Zhang et al. (Zhang et al. 2023b) design a multi-task learning framework to achieve medicine recommendation. In this work, we explore cross-category correlations with multi-task learning, demonstrating its effectiveness in multi-category warehouse demand prediction.

LMGL−WD Design

In this section, we first show the overview of LMGL−WD, followed by the detailed designs.

LMGL−WD Framework

As illustrated in Figure 1, LMGL−WD includes three components for category-level warehouse demand prediction:

- **LLM-Guided Category Series Encoding** obtains each category representation in a single warehouse with a category-level contextual encoder and a category-level series encoder;
- **Cross-Warehouse Category Learning** mines informative knowledge from cross-warehouses to complement the category representation effectively;
- **Cross-Category Multi-Task Learning** adaptively captures the intricate correlation across various categories for multi-category warehouse demand co-prediction.

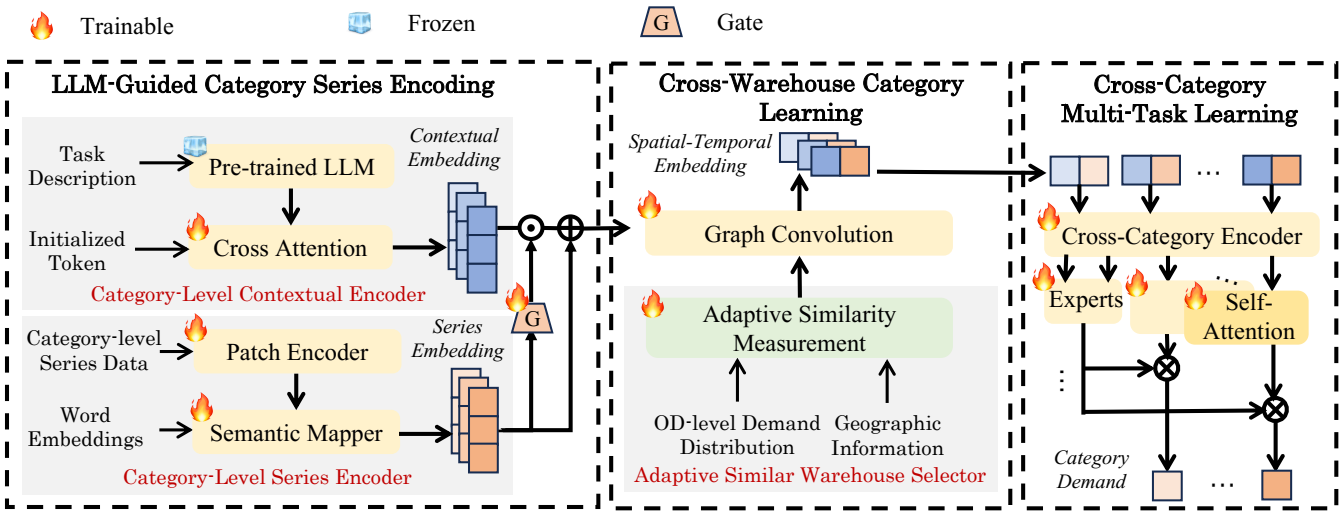

Figure 1: The Framework of LMGL-WD.

LLM-Guided Category Series Encoding

In this section, we explore the LLM's strong capability in understanding warehouse-related and category-related textual descriptions and time series data to enhance the category representation in a single warehouse.

Category-Level Contextual Encoder We first design a prompt that incorporates multi-view detailed textual information, including dataset and task descriptions, warehouse and category details, as well as prior knowledge. Then, we propose a category-level contextual encoder to extract the most informative knowledge from the prompt adaptively. Specifically, we first feed the prompt into the pre-trained LLM to obtain a contextual embedding vector M.

After that, an initialized ⟨context⟩ token c is introduced to extract effective context from prompt features via the cross-attention design.

$$Q_c = c \cdot W_Q \tag{1}$$

$$K_c = M \cdot W_K \tag{2}$$

$$V_c = M \cdot W_V \tag{3}$$

$$C = \mathit{CrossAttention}(Q_c, K_c, V_c) \tag{4}$$

where W_Q, W_K, W_V are the parameters. The output contextual embedding C contains useful contextual knowledge from the prompt.

Category-Level Series Encoder To address the modality gap between time series data and its semantic representation, we adopt a reprogramming strategy (Jin et al. 2023) to achieve deep alignment of cross-modal features.

Patch Encoder: A warehouse i's demand data over past t days is denoted as $\mathcal{H}^i = \{X_t^i, X_{t-1}^i, \ldots, X_1^i\}$, where $X_\tau^i = \{c_1, c_2, \ldots, c_N\}$ contains the demand of N categories in warehouse i on day τ. For input data $\mathcal{H} \in \mathbb{R}^{B \times N \times T}$ (where B is the batch size, N is the number of categories, and T is the series length), to obtain the patch of the c-th category in the b-th sample, denoted as $Patch_k^{(b,c)}$, circular padding is first applied to address boundary effects, followed by partitioning using a window size p and stride s:

$$Patch_k^{(b,c)} = \mathcal{H}[b, c, k \cdot s : k \cdot s + p] \tag{5}$$

where $k = 1, 2, \ldots, M$ and $M = \lfloor (T - p)/s \rfloor + 1$ is the total number of patches. Each patch is mapped to the feature space via a 1-D convolutional network:

$$p_k^{(b,c)} = \mathit{Conv1D}(Patch_k^{(b,c)}) \tag{6}$$

Here, *Conv1D* uses a 3×1 convolution kernel with a $LeakyReLU$ activation function, ultimately forming the time series embedding matrix.

Semantic Mapper: To map time series patch features to the word embedding space of the pre-trained language model, a set of text prototypes is learned through linear probing from the pre-trained word embedding matrix, capturing language cues related to time series patterns. A multi-head cross-attention layer is then used to establish associations between local patch information and text prototypes. By aggregating outputs from all attention heads and performing linear projection, the series embedding T in the semantic space is obtained as follows:

$$T = \mathit{Concat}(Z_1, \ldots, Z_{HA}) \cdot W_o \tag{7}$$

where W_o is the output projection matrix. Z_i is output of the i-th attention head.

Although category-level series have been reprogrammed to the LLM dimension, they still lack task-specific awareness for particular prediction tasks. The contextual embedding C now contains effective contextual information, and the determination of which information is useful for time

series modeling can be autonomously obtained by the time-series features. Mean pooling is performed on T to obtain T', and a gate mechanism is applied to T' to learn dynamic weights for contextual embedding, thereby filtering out the context useful for category-level demand prediction:

$$T' = \textit{MeanPooling}(T) \tag{8}$$

$$W_3 = \sigma\left(W_2 \cdot \textit{ReLU}\left(W_1 \cdot T'\right)\right) \tag{9}$$

$$C' = W_3 \odot C \tag{10}$$

where σ denotes the *sigmoid* function, $\textit{MeanPooling}(\cdot)$ represents the average pooling operation applied to the time-series feature T, $\odot$ denotes element-wise multiplication, and W_1, W_2, W_3 are learnable weight matrices.

The final fused category representation is:

$$T_{emb} = C' + T \tag{11}$$

Cross-Warehouse Category Learning

Intuitively, the warehouses with similar features can provide effective information to enhance the category demand prediction in the target warehouse. Relying on static geographical features or administrative affiliations to construct similar warehouse graphs has struggled to capture the dynamically changing characteristics of warehouse demand. To solve this limitation, we propose an adaptive similarity measurement scheme to construct the warehouse graph adaptively.

Adaptive Similarity Measurement For warehouse i, we obtain the distribution features by calculating the mean and standard deviation of demand data $\hat{H}^i$ over the last t days as $\hat{\mathcal{H}}^i = \{X^i_{m,t}, X^i_{s,t}, \ldots, X^i_{m,1}, X^i_{s,1}\}$, where $X^i_{m,\tau}$ and $X^i_{s,\tau}$ represent the mean value and standard deviation value of demands across categories on day τ, respectively. By compiling the daily average total demand from warehouse i to different geographical regions (i.e., East China, South China, North China, Central China, Southwest China, Northwest China, Northeast China) and cities of different economic tiers (i.e., from tier 1 to tier 5) from historical data, the destination distribution features $\mathcal{D}_i = \{d_1, d_2, ..., d_{Num}\}$ are formed (where Num is the total number of region-city tier combinations). $d_k \in \mathcal{D}_i$ denotes the average total demand from warehouse i to the k-th type of combination. The edge weight between warehouse i and j is calculated through a learnable function that fuses the two different types of features:

$$W_{i,j} = \sigma\left(\alpha(\hat{\mathcal{H}}^i \oplus \hat{\mathcal{H}}^j) + \beta(\mathcal{D}_i \oplus \mathcal{D}_j)\right) \tag{12}$$

where α and β are learnable parameter vectors, $\oplus$ denotes the feature concatenation operation, and the *sigmoid* function σ normalizes the weight to the interval $[0, 1]$ to quantify the similarity strength between warehouses. Through this method, we construct a fully connected graph containing all warehouse nodes, where there is an edge with weight $W_{i,j}$ between any two warehouses.

Graph Convolution To reduce computational complexity and focus on key association information, we adopt a Top-K subgraph strategy to filter important nodes and combine them with graph convolutional networks for feature aggregation. For the central warehouse i, based on the edge weights $W_{i,j}$ ($j \neq i$), the top K warehouses with the highest similarity score are selected, forming a subgraph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ containing $K+1$ nodes. The node set $\mathcal{V}_i = \{i\} \cup \{j_1, j_2, \ldots, j_K\}$, where $j_1, \ldots, j_K$ are the top-K similar warehouses. The edge set $\mathcal{E}_i$ of the subgraph includes connections between the central warehouse and the top-K warehouses, i.e., $\mathcal{E}_i = \{(i, j_k), (j_k, i) | k = 1, \ldots, K\}$, with edge weights W_{i,j_k}.

After the subgraph construction, a graph convolutional network (GCN) is used to aggregate features of the selected subgraph. The GCN maps the original features of subgraph nodes to the hidden space:

$$h_u^{(1)} = \textit{ReLU}\left(W_4 \cdot \sum_{v \in \mathcal{N}(u)} W_{u,v} \cdot {T_{\text{emb}}}^v\right) \tag{13}$$

where $\mathcal{N}(u)$ is the neighbor set of node u in the subgraph, T^v_{emb} is the output of the shipment data of warehouse v over the past t days after history embedding, W_4 is the weight matrix. The final graph feature of the central warehouse is obtained through aggregation with a linear layer. Specifically, the hidden features of all nodes in the subgraph are first concatenated along the feature dimension to form a high-dimensional vector, which is then normalized and projected through a linear transformation:

$$G^i_{emb} = W_5 \cdot \textit{LayerNorm}\left(\left[h_i^{(1)}; h_{j_1}^{(1)}; \ldots; h_{j_K}^{(1)}\right]\right) \tag{14}$$

where W_5 is the output projection matrix, *LayerNorm* denotes the layer normalization operation, and $[\cdot;\cdot;\ldots;\cdot]$ represents the feature concatenation operation. The resulting graph embedding of warehouse i is represented as G^i_{emb}, encapsulating the aggregated structural information of the subgraph centered at the warehouse i.

Cross-Category Multi-Task Learning

We design a cross-category multi-task learning module to fully explore the intrinsic correlations across different categories. This module can adaptively capture the relevant information from other categories through a self-attention mechanism to improve the targeted category demand prediction. Specifically, the graph embedding vector G^i_{emb} of a central warehouse i is first split into N independent feature vectors $\{g_i\}_{i=1}^N$ along the category dimension, which are then fed into their corresponding category-specific prediction experts.

Given the expert parameters W_i^N, the initial prediction result for a single category is:

$$\hat{y}_i^{(0)} = g_i \cdot W_i^N \tag{15}$$

Then, we concatenate the results of each expert to obtain the output:

$$\hat{Y}^{(0)} = \textit{Concat}(\hat{y}_1^{(0)}, \hat{y}_2^{(0)}, \ldots, \hat{y}_N^{(0)}) \tag{16}$$

Subsequently, using the graph embedding output vector, we learn the demand similarity relationships between categories via self-attention:

$$A = \textit{Softmax}\left(\frac{G^i_{\text{emb}} \cdot (G^i_{\text{emb}})^\top}{\sqrt{D}}\right) \tag{17}$$

where D is the feature dimension. This matrix facilitates the extraction of demand similarity patterns across categories.

After that, we obtain the final demand prediction results for all categories, considering the correlation information among categories:

$$\hat{Y} = A \cdot (\hat{Y}^{(0)})^\top \tag{18}$$

where $\hat{Y}$ is the final predicted demand value.

Model Training

In LMGL−WD, we adopt the Mean Absolute Percentage Error (MAPE) as the loss function:

$$L = \sum_{n=1}^{N} w_n \cdot \left|\frac{Y_n - \hat{Y}_n}{Y_n}\right| \tag{19}$$

Y_n and $\hat{Y}_n$ are the true and predicted values of category n, respectively. Meanwhile, due to the differences in the scales of different categories, we add w_c to represent the weight of each category.

Evaluation

In this section, we first describe the data and the experimental settings. After that, we show the extensive evaluation results to answer the following four research questions:

- *RQ1*: How does LMGL−WD perform compared to state-of-the-art methods under various input and output length settings?
- *RQ2*: How does LMGL−WD perform on different category demand prediction?
- *RQ3*: How does LMGL−WD perform in different cities?
- *RQ4*: How does LMGL−WD perform in zero-shot prediction scenarios?

Data and Metric

Dataset We collect order-related data from the warehouses of 120 cities in one of the largest e-commerce platforms in China. The order data involves 24 categories in 8 months. We utilize data on the total distribution of orders from each warehouse to each destination region over 45 days to construct correlations between warehouses.

Evaluation Metrics We adopt Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to evaluate the performance of different demand prediction methods.

Experimental Setup

Implementation Details The model training is conducted using 4 Ascend 910B NPUs, each with 64GB of memory, utilizing Python 3.10 and torch-npu 2.1.0. We use Llama-3-8B as the LLM, with training accelerated using DeepSpeed ZeRO-2. The batch size per NPU is set to 4, and the learning rate is 1e-3.

Baselines We select the following five state-of-the-art methods for comparison to evaluate the performance of LMGL−WD:

- **Autoformer** (Wu et al. 2021): This method focuses on the periodic and trend features of historical time series. It gradually separates trend and periodic components through latent variable decomposition, achieving progressive sequence decomposition.
- **DLinear** (Zeng et al. 2023): This is a simple yet efficient long-term time series forecasting structure, enabling future data prediction through a single-layer linear model.
- **iTransformer** (Liu et al. 2023): It is a transformer-based architecture, which treats time points as feature dimensions and sequences as tokens. It focuses on addressing channel independence in multivariate prediction and models temporal correlations across different time steps via attention mechanisms.
- **TimeLLM** (Jin et al. 2023): This is a time series prediction method based on large language models (LLMs). It reprograms time-series information into the feature space of LLMs and incorporates prior knowledge in natural language form to assist prediction.
- **TimeCMA** (Liu et al. 2024): It is an LLM-empowered encoding strategy, which packages time-series data into text prompts containing timestamps and numerical information to enhance prediction performance.

In addition, we also design two variants of LMGL−WD:

- LMGL−WD−G: The adaptive similar warehouse selector in LMGL−WD is removed.
- LMGL−WD−M: The cross-category multi-task learning component in LMGL−WD is removed.

Evaluation Results

In this section, we demonstrate the superiority of our category-level demand prediction framework LGML−WD through extensive experiments.

Overall Performance (RQ1) Table 1 presents the category-level warehouse demand prediction results of LMGL−WD and its variants, and five baselines under different scenarios. Specifically, we design nine prediction combinations, with the input series length from 7 to 21, and the output being the demand of each category in the next 3, 5, and 7 days. LMGL−WD achieves the best performance in two metrics, i.e., RMSE and MAPE, in all studied scenarios, compared to state-of-the-art methods. LMGL−WD also performs well in terms of MAE. Specifically, LMGL−WD reduces MAPE by 20.76%∼31.59%, and reduces RMSE by up to 37.7%. Two variants, i.e., LMGL−WD−G

Input Length	Methods	Predicted Next Days								
		3			5			7		
		RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
7	AutoFormer	2.691	1.305	27.10	3.206	1.556	27.79	3.624	1.699	28.06
	DLinear	2.674	1.314	26.89	3.201	1.547	27.17	3.621	1.705	27.46
	iTransformer	2.612	1.283	26.24	3.147	1.532	26.93	3.604	1.691	27.29
	TimeLLM	2.573	1.286	25.97	3.072	1.516	26.67	3.497	1.683	27.15
	TimeCMA	2.577	1.286	25.48	3.076	1.521	26.37	3.513	1.691	27.18
	LMGL-WD-G	2.172	1.241	21.61	2.915	1.522	21.61	3.225	**1.663**	23.47
	LMGL-WD-M	2.113	1.229	20.46	2.881	1.511	20.61	3.192	1.676	22.41
	LMGL-WD	**2.071**	**1.223**	**19.20**	**1.997**	**1.503**	**19.01**	**3.182**	1.682	**20.71**
14	AutoFormer	2.192	1.233	27.92	2.879	1.469	28.47	3.117	1.511	28.87
	DLinear	2.183	1.231	27.55	2.886	1.445	28.01	3.113	1.493	28.51
	iTransformer	2.158	1.148	27.33	2.839	1.432	27.69	3.101	1.435	28.48
	TimeLLM	2.901	1.117	27.01	2.828	1.314	27.79	2.997	**1.428**	28.33
	TimeCMA	2.312	1.147	26.13	2.824	1.288	27.04	3.012	1.441	27.71
	LMGL-WD-G	1.983	1.121	23.20	2.482	1.396	23.57	2.992	1.473	25.12
	LMGL-WD-M	1.962	**1.091**	21.84	2.452	1.315	22.51	**2.731**	1.508	23.89
	LMGL-WD	**1.872**	1.113	**20.36**	**1.872**	**1.278**	**20.96**	2.891	1.431	**21.55**
21	AutoFormer	1.773	1.158	28.71	1.969	1.327	29.31	2.707	1.409	29.66
	DLinear	1.756	1.187	28.47	1.957	1.308	28.89	2.701	1.401	29.37
	iTransformer	1.732	1.117	28.25	1.935	1.297	28.61	2.697	1.397	29.25
	TimeLLM	1.635	1.183	27.93	1.867	1.227	28.42	2.563	1.384	29.01
	TimeCMA	1.707	1.115	26.97	1.905	1.241	27.78	2.567	**1.383**	28.19
	LMGL-WD-G	1.631	1.156	24.28	1.881	1.283	24.51	2.599	1.397	25.15
	LMGL-WD-M	1.612	1.128	22.87	1.865	**1.183**	23.09	2.551	1.469	24.32
	LMGL-WD	**1.557**	**1.113**	**21.37**	**1.557**	1.223	**21.69**	**2.539**	1.381	**22.05**

Table 1: The Overall Performance. The Best Results are Bolded, and the Second-best Results are Underlined.

and LMGL-WD-M, perform slightly worse than LMGL-WD, demonstrating the effectiveness of the adaptive similar warehouse selector and cross-category multi-task learning components in LMGL-WD.

Category	RMSE	MAE	MAPE(%)
Bedding	**1.69**	**1.20**	**16.29**
Food & Beverages	2.11	1.37	19.62
Office Supplies	1.73	1.21	16.49
Beauty & Skincare	2.09	1.39	19.39
Pet Supplies	1.97	1.31	18.54
Alcohol	2.57	1.43	22.69

Table 2: The Impact of Categories. The Best Results are Bolded, and the Second-best Results are Underlined.

The Impact of Categories (RQ2) We further analyze the warehouse demand prediction performance of different categories. We use 7-day historical demand data to predict the demand for the next 3 days. The results of some typical categories are presented in Table 2. We selected 6 categories, i.e., Bedding, Food & Beverages, Office Supplies, Beauty & Skincare, Pet Supplies, and Alcohol, from the 24 available categories, and their demands decreased in that order. Among them, Bedding achieved the best prediction performance in all metrics, because of the stable daily demand. In contrast, Alcohol has a more volatile demand, resulting in relatively poor prediction performance.

The Impact of City Tiers (RQ3) We evaluate LMGL-WD across cities with different tiers to explore the impact of city tiers on the category-level warehouse demand prediction. We divided the studied cities into five tiers (Sun et al. 2024) based on complicated assessment criteria, and tier 1 and the new tier 1 cities are combined. The demand for warehouses decreases from tier 1 to tier 5 cities. As shown in Table 3, tier 1 and tier 2 cities achieve better performance in terms of MAPE in all scenarios, while tier 4 and tier 5 cities performed better in RMSE and MAE.

Zero-Shot Prediction (RQ4) In this section, we further evaluate the zero-shot prediction capability of LMGL-WD on cross-city data. Specifically, we divided the studied cities into the following three groups based on their tiers while ensuring each group contained the same number of cities: Group A (tier 1 and tier 2 cities), Group B (tier 3 and tier 4 cities), and Group C (tier 5 cities). We conduct cross-city tests using 7-day historical data to predict demand for the next 3 days. The training and testing processes are constructed as follows: Case 1 (training on Group A and testing on Group B), Case 2 (training on Group A and testing on

Input Length	City Tier	Predicted Next Days								
		3			5			7		
		RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
7	Tier 1	1.954	1.287	**11.64**	3.051	1.721	**14.31**	3.298	1.883	**14.97**
	Tier 2	2.265	1.351	13.86	2.512	1.571	15.42	3.477	1.692	15.72
	Tier 3	3.544	1.452	17.43	2.701	1.532	17.82	2.964	1.681	18.02
	Tier 4	**1.937**	1.239	21.85	2.578	1.462	21.87	2.588	**1.493**	25.49
	Tier 5	1.891	**1.203**	24.55	**1.968**	**1.287**	23.09	**2.312**	1.582	23.64
14	Tier 1	1.677	1.131	**12.43**	2.041	1.377	**16.14**	2.241	1.476	**13.74**
	Tier 2	1.867	1.251	14.32	2.267	1.420	17.55	3.293	1.537	16.76
	Tier 3	1.683	1.128	17.14	1.918	1.382	19.82	1.877	1.372	18.94
	Tier 4	**1.639**	1.121	25.61	2.053	**1.231**	23.07	1.940	1.308	24.42
	Tier 5	1.662	**1.073**	23.93	**1.865**	1.282	23.27	**1.743**	**1.225**	24.35
21	Tier 1	1.707	1.241	**16.72**	1.797	1.267	**15.64**	2.068	1.543	**18.37**
	Tier 2	1.463	1.130	18.15	1.841	1.382	16.04	3.166	1.690	18.85
	Tier 3	1.471	1.113	19.48	1.991	1.272	18.47	2.067	1.292	20.94
	Tier 4	1.459	1.001	23.25	1.752	1.251	24.12	**1.774**	1.317	25.12
	Tier 5	**1.424**	**0.988**	27.05	**1.741**	**1.237**	27.37	1.872	**1.288**	28.62

Table 3: The Impact of City Tiers. The Best Results are Bolded, and the Second-best Results are Underlined.

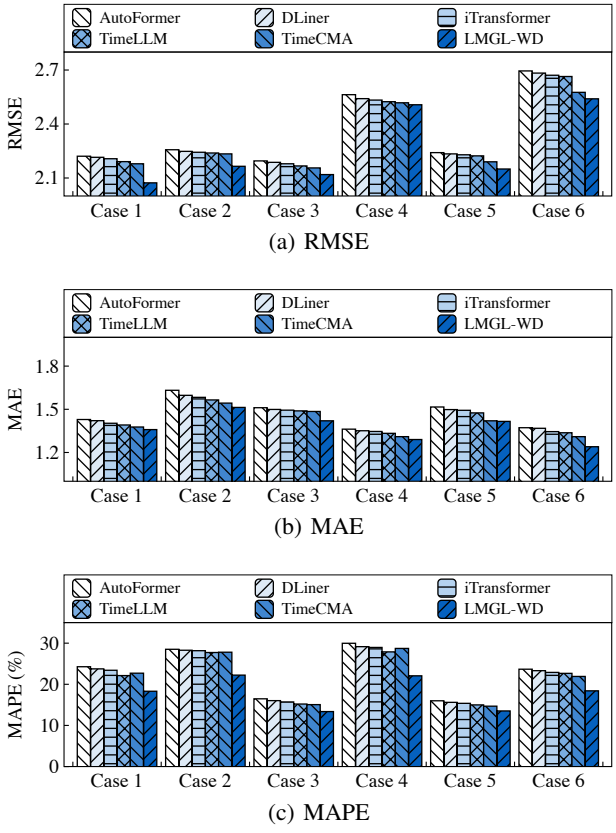

Figure 2: Zero-Shot Prediction Performance.

Group C), Case 3 (training on Group B and testing on Group A), Case 4 (training on Group B and testing on Group C), Case 5 (training on Group C and testing on Group A), and Case 6 (training on Group C and testing on Group B). The experimental results are presented in Figure 2. LMGL-WD exhibits promising zero-shot prediction capabilities. Specifically, LMGL-WD reduces RMSE, MAE, and MAPE by up to 3.08%, 23.18%, and 4.37%, respectively, compared to state-of-the-art methods.

Conclusion

In this paper, we propose an LLM-guided multi-task graph learning framework, LMGL-WD, for category-level warehouse demand prediction in e-commerce. We first design an LLM-guided category series encoding module to represent each category based on contextual and series embeddings. Then, a cross-warehouse category learning module is designed to enhance the category representation by effectively capturing the informative cross-warehouse knowledge. Finally, we design a cross-category multi-task learning module to adaptively capture the cross-category correlations for cooperative demand prediction of various categories. Extensive evaluation results with real-world data collected from the largest warehouse-based e-commerce company in China demonstrate that LMGL-WD achieves better performance, e.g., reduces MAPE by up to 31.59%, compared to state-of-the-art methods.

Acknowledgments

This work is supported partly by the National Natural Science Foundation of China (NSFC) 62576013, National Key Research Plan under grant No.2024YFC2607404, the Jiangsu Provincial Key Research and Development Program under Grant BE2022065-1, BE2022065-3, and the Ningxia Domain-Specific Large Model Health Industry R&D No. 2024JBGS001.

References

Amazon. 2025. Amazon Official Website.

Bandara, K.; Shi, P.; Bergmeir, C.; Hewamalage, H.; Tran, Q.; and Seaman, B. 2019. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *International conference on neural information processing*, 462–474. Springer.

Cai, T.; Wan, H.; Wu, F.; Wen, H.; Guo, S.; Wu, L.; Hu, H.; and Lin, Y. 2023a. M^2g4rtp: 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.

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023b. LightGCL: Simple yet effective graph contrastive learning for recommendation. *arXiv preprint arXiv:2302.08191*.

Cao, D.; Jia, F.; Arik, S. O.; Pfister, T.; Zheng, Y.; Ye, W.; and Liu, Y. 2023. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*.

Challu, C.; Olivares, K. G.; Oreshkin, B. N.; Ramirez, F. G.; Canseco, M. M.; and Dubrawski, A. 2023. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 6989–6997.

Feng, X.; Zhong, S.; Hang, J.; Lyu, W.; Zhang, Y.; Yang, G.; Wang, H.; Zhang, D.; and Wang, G. 2025. Hierarchical Structure Sharing Empowers Multi-task Heterogeneous GNNs for Customer Expansion. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 4424–4434.

Fu, L.; Deng, B.; Huang, S.; Liao, T.; Zhang, C.; and Chen, C. 2025. Learn from Global Rather Than Local: Consistent Context-Aware Representation Learning for Multi-View Graph Clustering. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, 16–22.

Guo, B.; Song, X.; Wang, S.; Gong, W.; He, T.; and Liu, X. 2024. Multi-task Conditional Attention Network for Conversion Prediction in Logistics Advertising. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5028–5037.

Hu, Y.; Wang, X.; Ren, H.; He, T.; Tang, T.; He, H.; Meng, C.; Han, B.; Bao, J.; Sun, Y.; et al. 2023. Epidemic Amplifier Detection: Finding High-Risk Locations in COVID-19 Cases' Location Sequences via Multi-task Learning. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 1–4.

Huang, S.; Fu, L.; Zhuang, S.; Qiu, Y.; Huang, B.; Cui, Z.; and Zhang, T. 2025a. Going beyond consistency: target-oriented multi-view graph neural network. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 5426–5434.

Huang, W.; Liang, J.; Wan, G.; Zhu, D.; Li, H.; Shao, J.; Ye, M.; Du, B.; and Tao, D. 2025b. Be confident: Uncovering overfitting in mllm multi-task tuning. In *Forty-second International Conference on Machine Learning*.

JD-Retail. 2025. JD Retail Official Website.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; et al. 2023. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*.

Jin, M.; Zhang, Y.; Chen, W.; Zhang, K.; Liang, Y.; Yang, B.; Wang, J.; Pan, S.; and Wen, Q. 2024. Position: What can large language models tell us about time series analysis. In *41st International Conference on Machine Learning*. MLResearchPress.

Kong, L.; Yang, H.; Li, W.; Zhang, Y.; Guan, J.; and Zhou, S. 2024. Traffexplainer: A Framework Toward GNN-Based Interpretable Traffic Prediction. *IEEE Transactions on Artificial Intelligence*, 6(3): 559–573.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2024. Timecma: Towards llm-empowered time series forecasting via cross-modality alignment. *arXiv e-prints*, arXiv–2406.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Pan, Z.; Jiang, Y.; Garg, S.; Schneider, A.; Nevmyvaka, Y.; and Song, D. 2024. s^2 IP-LLM: Semantic space informed prompt learning with LLM for time series forecasting. In *Forty-first International Conference on Machine Learning*.

Qi, Y.; Li, C.; Deng, H.; Cai, M.; Qi, Y.; and Deng, Y. 2019. A deep neural framework for sales forecasting in e-commerce. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 299–308.

Shi, J.; Yao, H.; Wu, X.; Li, T.; Lin, Z.; Wang, T.; and Zhao, B. 2021. Relation-aware meta-learning for e-commerce market segment demand prediction with limited records. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 220–228.

Singh, B.; Kumar, P.; Sharma, N.; and Sharma, K. 2020. Sales forecast for amazon sales with time series modeling. In *2020 first international conference on power, control and computing technologies (ICPC2T)*, 38–43. IEEE.

Sun, C.; Li, H.; Li, Y.; and Hong, S. 2023. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv preprint arXiv:2308.08241*.

Sun, Y.; Zhu, H.; Wang, L.; Zhang, L.; and Xiong, H. 2024. Large-scale online job search behaviors reveal labor market shifts amid COVID-19. *Nature Cities*, 1(2): 150–163.

Wang, B.; Chen, J.; Li, C.; Zhou, S.; Shi, Q.; Gao, Y.; Feng, Y.; Chen, C.; and Wang, C. 2024. Distributionally robust graph-based recommendation system. In *Proceedings of the ACM web conference 2024*, 3777–3788.

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.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34: 22419–22430.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Yang, G.; Tu, M.; Li, Z.; Hang, J.; Liu, T.; Liu, R.; Ding, Y.; Yang, Y.; and Zhang, D. 2024. Behavior-Aware Hypergraph Convolutional Network for Illegal Parking Prediction with Multi-Source Contextual Information. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2827–2836.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

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. 2023a. 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.

Zhang, Y.; Wu, X.; Fang, Q.; Qian, S.; and Xu, C. 2023b. Knowledge-enhanced attributed multi-task learning for medicine recommendation. *ACM Transactions on Information Systems*, 41(1): 1–24.

Zhong, S.; Liu, K.; Lyu, W.; Wang, H.; Wang, G.; Liu, Y.; He, T.; Yang, Y.; and Zhang, D. 2025. Adaptive Multi-Faceted Service Capabilities Co-Prediction for Nationwide Terminal Stations in Logistics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 28557–28565.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, 27268–27286. PMLR.