Peak Period Demand Forecasting with Proxy Data: GNN-Enhanced Meta-Learning

Zexing Xu[1], Linjun Zhang[2], Sitan Yang[3], Nan Jiang[4]
[1]Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign
[2]Department of Statistics, Rutgers University
[3]Forecasting Science, Amazon
[4]Department of Computer Science, Purdue University
zexingx2@illinois.edu, linjun.zhang@rutgers.edu, sitanyan@amazon.com, jiang631@purdue.edu

Abstract

Demand prediction is a crucial task for e-commerce and physical retail businesses, especially during high-stake sales events. However, the limited availability of historical data from these peak periods poses a significant challenge for traditional forecasting methods. In this paper, we propose a novel approach that leverages proxy data from non-peak periods, enriched by features learned from a graph neural networks (GNNs) based forecasting model, to predict demand during peak events. We formulate the demand prediction as a meta-learning problem and introduce the Feature-based First-Order Model-Agnostic Meta-Learning (F-FOMAML) algorithm, which adapts to new tasks by conditioning on the GNN-generated relational metadata. Empirical evaluations on large-scale industrial datasets demonstrate the superiority of our approach, with our model consistently outperforming state-of-the-art baselines in the demand prediction task, by 26.24% *on the internal vending machine dataset and* 8.7% *on the public JD.com dataset over the Mean Absolute Error.*

1. Introduction

Forecasting product demand during high-stake sales events such as Black Friday or Prime Day is a daunting task for both e-commerce giants like Amazon and JD.com and physical retailers. This challenge stems largely from the scarcity of event-specific historical data. Commonly, businesses anchor their strategies to regular sales data, which may not fully capture the distinct consumer behaviors observed during promotional periods. Beyond standard demand prediction, promotional forecasting includes predicting "extreme" events [19]. These events, marked by deeper discounts and atypical merchandising strategies, significantly deviate from the typical sales patterns influenced by factors like seasonality or product life-cycles. This deviation necessitates a specialized approach to deal-level forecasting, one that thoroughly considers promotion-specific intricacies, from the depth of discounts to deal combinations.

For instance, an online retailer aiming to anticipate the demand spike for a newly launched electronic item during a holiday sale might struggle. They might be unsure how various promotions will influence demand during these events, particularly when previous similar event data is limited or non-existent.

To mitigate this, our research makes effective use of proxy data from non-peak sales to inform decisions during peak sales events. However, this supplemental data alone is insufficient, given the intricate inter-relationships among various products, categories, and even across different shopping platforms. We thus introduce a representation learning task for each product, leveraging a cutting-edge Graph Neural Network (GNN) based forecasting model [44]. This model generates embeddings, enriched with graph-enhanced features, encapsulating cross-product information derived from pertinent graph structures. Such structures offer insights into a myriad of dynamics, from relationships between products to patterns of inter-platform shopping behaviors.

At the heart of our methodology is the Feature-based First-Order Model-Agnostic Meta-Learning (F-FOMAML) approach. This approach refines the foundational MAML framework [11, 27], incorporating task-specific insights drawn from GNN-processed data. By training F-FOMAML with this enhanced metadata, the model showcases an unparalleled ability to adapt, consistently surpassing conventional forecasting techniques in accuracy metrics.

While we primarily target enhancing e-commerce and brick-and-mortar retail demand prediction, the potential of our GNN-augmented F-FOMAML is vast. Its versatility

makes it a candidate for various applications, from fortifying online banking fraud detection systems to optimizing digital advertising click-through rates.

To summarize, our main contributions are:

1. *Model:* We propose a novel approach to model demand prediction, reframing it as a graph-augmented meta-learning challenge.

2. *Algorithm:* We introduce the GNN-infused F-FOMAML algorithm, which skillfully combines meta-learning and the feature-wise linear modulation (FiLM) layers. This results in a model capable of producing robust predictions, even when historical data is sparse.

3. *Numerical Experiments:* Empirical tests validate F-FOMAML's proficiency, with the model consistently outshining existing forecasting methods in the prediction MAE values, by 26.24% on the vending machine dataset and 8.7% on the JD.com dataset using domain-knowledge constructed features. Furthermore, our model achieves an 11.44% improvement over the MAE metric against baselines with GNN integrated.

2. Related Work

Prediction with Limited Data Previous work in transfer learning has focused on learning from data-rich domains and transferring knowledge to data-sparse regions or underrepresented classes [15, 16]. In the context of vending machine merchandising, we aim to learn from well-performing vending machines to improve the performance of underperforming or newly installed machines. Multi-task learning has also been used to enhance model performance on data-sparse tasks [2, 6].

To benefit the learning process on tasks with limited data, transferring knowledge from related tasks has achieved great success in recent years [29]. Conventional transfer learning methods learn transferable latent factors between one source domain and one target domain. The latent factors are captured by a series of techniques, such as matrix factorization [25], manifold learning [13], and deep learning [24, 39]. In our work, we focus on adopting meta-learning techniques to learn from various tasks and then adapt them to unseen tasks in the context of vending machine merchandising.

Meta-Learning for Demand Prediction Meta-learning has been applied to various retail and demand prediction tasks, with an emphasis on learning from diverse data sources and adapting to new tasks with limited data. For instance, [21] employed meta-learning to predict demand in retail settings, demonstrating the effectiveness of meta-learning in capturing complex patterns across diverse scenarios. Similarly, [41] applied meta-learning to online retail data, highlighting the potential for meta-learning in e-commerce applications.

In the time series-related problems, [28] briefly discusses the relation between neural time series prediction and meta-learning. [45] incorporate the gradient-based meta-learning with a region functionality based memory [45] for spatiotemporal prediction. However, this method relies on the spatial semantic correlations between tasks, which limits its applicability to our problem.

Our work relates and contributes to the problem of learning customer demand for new products with few historical data. Previous works have suggested comparing the features of new products to existing ones [1, 10], or efficient methods for eliciting additional information [5, 26]. Our paper assumes that sales have already been observed at a single price and leverages more information from other related products and vending machines as proxy data.

Meta-Learning Methods for Few-Shot Learning Existing works on meta-learning methods for few-shot learning can be categorized into metric-learning-based approaches that aim to establish similarity or dissimilarity between classes [36, 37, 40], and optimization-based approaches that seek to learn a good initialization point that can adapt to new tasks within a few parameter updates [11, 27, 31].

Graph Neural Networks for Time Series Forecasting Deep learning models have been extensively explored for time series forecasting especially those with the Seq2Seq structure [38], which involves learning an encoder to transform various inputs into fixed length hidden states for producing forecasts. Recent developments include DeepAR [32], TFT [23] and MQ-Forecasters [9, 42]. However these methods do not account for cross-observation information, which becomes important in many practical applications. As a result, Graph Neural Networks (GNNs) have rapidly emerged as a promising framework to address this issue by combining temporal processing with graph convolution to augment the learning of individual time series [18, 22, 33, 43]. A popular family of methods propose graph structure learning for the joint inference of a latent structure through GNN while forecasting [17, 43]. However they suffer limitations of scaling to large datasets. A scalable approach recently introduced by Yang et al. [44] uses predefined graphs as data augmentations rather than enabling graph structure learning, which demonstrates not only to scale to graphs over millions of nodes but also shows substantially improving model performance especially for cold-start problems when data is scarce.

Our work builds upon these foundations by specifically applying meta-learning and few-shot learning techniques to the demand forecasting problem, with the goal of improving the adaptability and performance of models in this context. To the best of our knowledge, we are the first to study peak period demand prediction with limited records by borrow-

ing relation-aware knowledge from other time period. We focus on this domain, exploring the application of meta-learning for few-shot prediction and the incorporation of auxiliary information, such as proxy data from other related tasks, to improve model performance.

3. Problem Formulation

We begin by framing our research problem in a generic setting, focusing on few-shot supervised learning within the meta-learning paradigm. We particularly focus on demand forecasting.

3.1. Task Definition

Demand forecasting aims to predict the future demand for a product in a specific environment based on observed features. Each forecasting task is associated with a product and its environment.

Formally, let $\mathcal{P}(\mathcal{T})$ denote a distribution over tasks $\mathcal{T}_{ij}$, each corresponding to product i in environment j. For a set of n products present in t_i environments, we have $\sum_{i=1}^{n} t_i$ tasks. Each task dataset is symbolized as a pair $(\mathbf{x}_{ij}, y_{ij})$, where $\mathbf{x}_{ij}$ is the feature vector and y_{ij} signifies the associated demand.

To provide a concrete example, consider a scenario where we have $n = 10$ products, each available in $t_i = 5$ locations. Therefore, we have a total of 50 tasks in our meta-training set. The dataset corresponding to each task is represented as a demand-feature pair $(\mathbf{x}_{ij}, y_{ij})$, where $\mathbf{x}_{ij} \in \mathbb{R}^m$ is a feature vector and $y_{ij} \in \mathbb{R}$ is the associated demand.

Our goal is to train a model, denoted $f : \mathbb{R}^m \to \mathbb{R}^+$, capable of mapping m-dimensional observations $\mathbf{x}$ to outputs y across a large or possibly infinite number of tasks. We employ the First-Order Model Agnostic Meta-Learning (FOMAML) algorithm for this purpose. For a given product characterized by a feature vector $s_i, \forall i \in [n]$ and an environment (e.g., location) characterized by a feature vector $v_j, \forall j \in [t_i]$, we consider a single historical price and demand observation $(\tilde{p}_{ij}, \tilde{y}_{ij})$.

Given a price of interest p_{ij}, we assume our task as the following demand function:

$$y_{ij} = f_{ij}(\mathbf{x}_{ij}) + \epsilon_{ij}, \tag{1}$$

where $\mathbf{x}_{ij}$ is the feature tuple $(s_i, v_j, \tilde{p}_{ij}, \tilde{y}_{ij}, p_{ij})$ and y_{ij} is the corresponding demand y_{ij}. Here, f_{ij} is a flexible function (e.g., linear regression, MLP, etc.) and each task is associated with a unique model parameter $\beta_{ij} \in \mathbb{R}^m$. We assume that the noise ϵ_{ij} follows a centered sub-Gaussian distribution with parameter σ_i^2. Furthermore, without loss of generality, we assume that $\mathcal{P}_X$ is an isotropic-centered sub-Gaussian distribution, i.e. $\mathbb{E}(\mathbf{x}_{ij}\mathbf{x}_{ij}^\top) = \mathbb{I}_d$.

Remark 1. *Incorporating features allows us to capture an additional form of shared structure. However, despite accounting for observed product features, the demand functions of two products can exhibit distinct behaviors. For instance, even for Diet Coke, price sensitivities may vary significantly on different vending machines due to factors such as customer demographics or preferences that are challenging to capture as explicit features. To account for these product-location-specific nuances, we introduce the flexibility for the demand function's coefficients (e.g., price elasticity) to differ.*

Exploiting some structural similarities in $\mathcal{P}(\mathcal{T})$, the goal is to train a model for a new task $\mathcal{T}^{\text{new}}$, coming from $\mathcal{P}(\mathcal{T})$, from a small amount of *training dataset* $\mathcal{D} = \left(\mathbf{x}_{ij}^{\text{new}}, y_{ij}^{\text{new}}\right)$.

In the First-Order MAML (FOMAML) approach, the model parameters for each task in the meta-training dataset are computed after a single gradient update. Specifically, for each task $\mathcal{T}_{ij}$, the task-specific model parameters, denoted β'_{ij}, are updated as follows:

$$\beta'_{ij} \leftarrow \beta^* - \lambda \nabla_{\beta^*} \mathcal{L}_{\mathcal{T}_{ij}}(\beta_{ij}), \tag{2}$$

where λ is the learning rate, β^* is the global model parameter shared across tasks, and $\mathcal{L}_{\mathcal{T}_{ij}}(\beta_{ij})$ is the task-specific loss, such as the mean squared error:

$$\mathcal{L}_{\mathcal{T}_{ij}}(\beta_{ij}) = \frac{1}{2}\left(y_{ij} - f_{ij}(\mathbf{x}_{ij})\right)^2. \tag{3}$$

After updating the task-specific parameters, a meta-update is performed on the shared global parameter β^* using the performance of the updated β'_{ij} on their corresponding tasks. This meta-update is given by the following:

$$\beta^* \leftarrow \beta^* - \eta \sum_{i=1}^{n} \sum_{j=1}^{t_i} \nabla_{\beta^*} \mathcal{L}_{\mathcal{T}_{ij}}(\beta'_{ij}), \tag{4}$$

where η is the meta-learning rate. The objective of this meta-learning process is to optimize the shared global parameter β^* such that, after a few updates on each individual task, the task-specific parameters β_{ij} yield improved performance on their corresponding tasks.

Once the meta-learning process is complete, the model parameters of a newly arriving task can be estimated using the learned meta-parameters β^*. These task-specific parameters β'_{ij} can then be fine-tuned on the new task using the available data, yielding improved performance and adaptability to new tasks.

By incorporating the FOMAML algorithm into our meta-learning framework, we aim to construct an efficient model for sales prediction that can swiftly adapt to new tasks with limited historical sales data.

4. Methodology

We illustrate the pipeline of our algorithm in Figure 1. Imagine there are 3 locations offering 6 drink types with

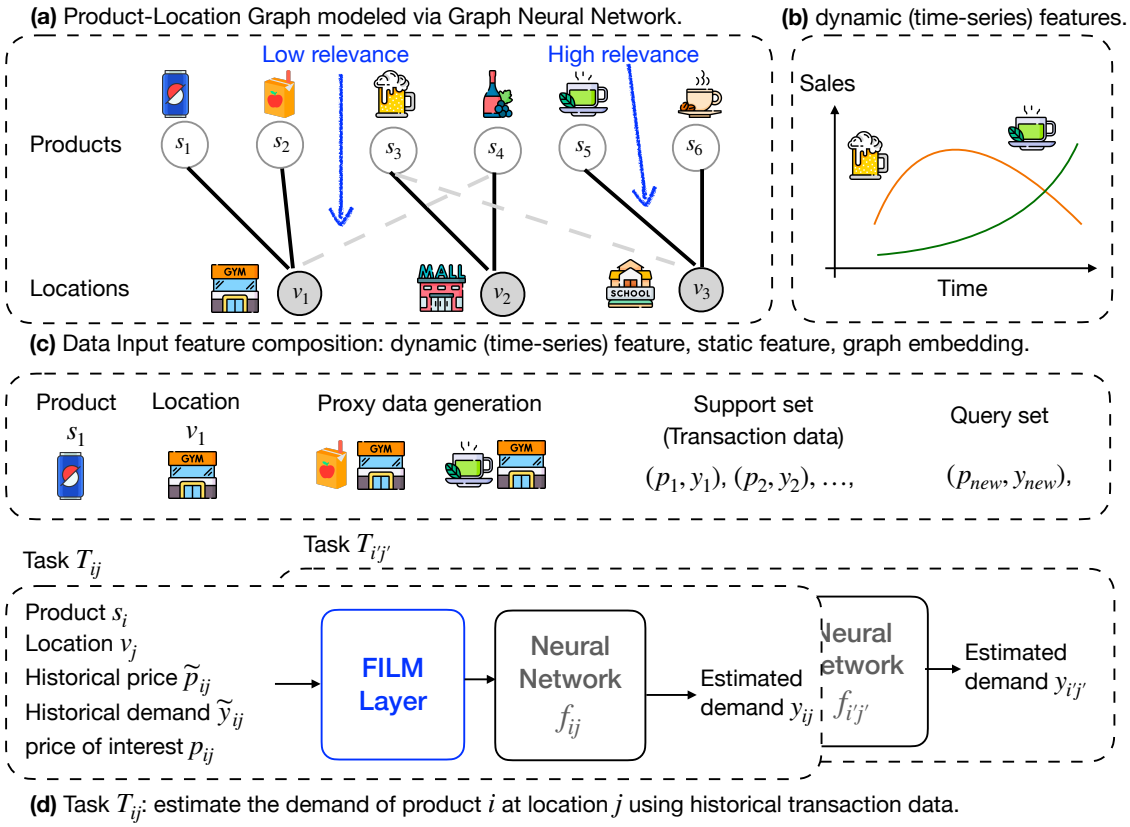

Figure 1. Pipeline of GNN-enhanced F-FOMAML for demand forecasting.

transaction data, capturing their historical sales. First, a graph neural network (GNN), G, is formed using both static features like machine locations and dynamic features from past sales time series. To predict the demand for coke at the gym with a discount, relevant nodes and edges from G are extracted. This subset, denoted as $G_{\mathcal{T}}$, undergoes training using MAML's inner loop, yielding initial task-specific parameters. These parameters are further refined through the FiLM transformer, considering proxy data that might suggest a promotion for Coke. The shared meta-parameters are updated in MAML's outer loop based on the specific task losses. Once this cycle is completed across all tasks, the model is evaluated on fresh data to project the demand.

Our proposed methodology for e-commerce demand prediction encompasses three pivotal components: proxy data selection, neural architecture determination, and the F-FOMAML algorithm design. To cater to the multi-faceted nature of e-commerce products and their varied demand across different locations or customer segments, we weave price-sensitivity estimators into the meta-learning framework. Further, we employ GNN and the FiLM layer to utilize and encode proxy data into hidden representations, thus enabling the modulation of learner parameters for enhanced adaptation to the specific characteristics of products and customer segments.

4.1. Proxy Data Selection

The proxy data, vital for tasks with limited historical sales data, is judiciously selected. The ideal proxy data simulates the potential sales behavior of the focal product, informed by sales trends of similar products or those in related categories.

For e-commerce scenarios, task similarity might arise from: 1) *Historical Transactions*: Edges represent products often purchased together. 2) *User Behavioral Patterns*: Edges might indicate similar purchase behaviors or browsing patterns of users. 3) *Product Similarities*: Linking products of the same category or with similar attributes. 4) *Domain Knowledge*: Connections deriving from expert insights into customer behaviors, seasonal trends, or market dynamics.

To quantify this resemblance, we employ clustering methods on the feature vectors $\mathbf{x}_{ij}$, such as k-means and hierarchical clustering. Distances are gauged using metrics like the Euclidean and cosine similarity. Then, for a given task, we denote its proxy data as Z_{ij}.

Graph Construction for Proxy Data A tailored graph for our GNN encapsulates relationships among tasks. In determining proxy data for e-commerce settings, we choose tasks from support set $\mathcal{T}$ resembling our target task, $\mathcal{T}_{\text{new}}$,

guided by:

$$\text{correlation}(\mathcal{T}, \mathcal{T}_{\text{new}}) > \delta \tag{5}$$

where δ is a threshold indicating task similarity.

4.2. GNN-enhanced Representation Learning

This section describes how we obtain the graph-enhanced features for each product. In a nutshell, we set up a time series forecasting task and utilize a GNN-based demand forecasting model to predict future sales given each product's historical information as well as cross-product relationships defined by a predefined graph. The trained model then produces the product embeddings as features.

Input Product Features E-commerce platforms host a plethora of products, each with unique characteristics and consumer interactions. In this case, We construct the graph using product-specific attributes such as brands. The input features for node N_i (representing product i) are: 1) Static features S_i, like the product category, brand, and manufacturing details. 2) Dynamic features D_i, encompassing time-evolving aspects like recent sales and price changes.

Product Embedding Generation via Forecasting A crucial aspect is to generate meaningful product embeddings that can capture the multifaceted nature of e-commerce products. To facilitate this, we set up a demand forecasting task as:

$$\widehat{Y}_{t+1} = f\left(Y_{t-C:t}, D_{t-C:t}, S\right), \tag{6}$$

Here f represents the forecasting model. At time t, target $Y_{t+1} \in \mathbb{R}^{N\times 1}$ are the future one-day sales, $D_{t-C:t} \in \mathbb{R}^{N\times d}$ are d dynamic features with the history length of C days and $S \in \mathbb{R}^{N\times m}$ are m static features for all N products. We adopt the GNN-based forecasting model introduced in [44] and we use brand information to craft the predefined graph. After training convergence, we extract the embeddings for each product, which serve as compact representations of product dynamics.

Edge Relationship Determination Let $E(\mathcal{T}_i, \mathcal{T}_j)$ denotes the edge between task $\mathcal{T}_i$ and $\mathcal{T}_j$. The edge relationships between the two entities are inferred using:

$$E(\mathcal{T}_i, \mathcal{T}_j) = \begin{cases} 1 & \text{if dist}(emb(\mathcal{T}_i), emb(\mathcal{T}_j)) < \theta \\ 1 & \text{if } h_{\mathcal{T}_i} = h_{\mathcal{T}_j} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $emb(\mathcal{T}_i)$ stands for the embedding of task $\mathcal{T}_i$, $\text{dist}(\cdot,\cdot)$ denotes a function measuring the distance between two embeddings, h denotes the task (i.e., product) hierarchy or taxonomy, and θ is a pre-determined threshold to determine closeness. We will create an edge between task $\mathcal{T}_i$ and $\mathcal{T}_j$ if either their corresponding embedding are close to each other or they belong to the same category.

4.3. F-FOMAML Algorithm Description

We present the GNN-Integrated First-Order MAML for Demand Forecasting, which incorporates transaction data, static and dynamic features, and proxy data to forecast demand. Pivoting on a variant of the MAML algorithm, our method delineates several stages of the learning journey: meta-learner, base learners, FiLM Layer and an ensemble with fine-tuning.

FiLM Layer The feature-wise linear modulation (FiLM) layer [30] is a critical component in tailoring the learner parameters based on the proxy data features. This layer applies an affine transformation, feature-wise, to its input, modulating the hidden vector outputs of the meta-model using the proxy data Z_{ij} as task encodings. The construction and purpose of the proxy data Z_{ij} is elaborated in section 4.1. The FiLM layer facilitates a more refined adaptation to the distinctive traits of the product and vending machine location by exploiting the relationship between the product-specific and machine-specific price-sensitivity estimators encapsulated in the proxy data.

The FiLM layer's mechanism can be mathematically depicted as:

$$\text{FiLM}(\mathbf{x}_{ij}) = \gamma(Z_{ij}) \odot \mathbf{x}_{ij} + \beta(Z_{ij}), \tag{8}$$

where X represents the input feature representation, $\gamma(Z_{ij})$ and $\beta(Z_{ij})$ (abv. β_{ij}) signify the scaling and shifting factors respectively learned from the proxy data Z_{ij}, and $\odot$ symbolizes element-wise multiplication. These functions, $\gamma(Z_{ij})$ and $\beta(Z_{ij})$, are learned during the training phase to cater to the specific task at hand. By applying this transformation to the task-specific model parameters β'_{ij}, the model captures complex feature interactions and becomes better equipped to adapt to the specific characteristics of each unique product-vending machine pair.

Meta-Learner The core of the meta-learning approach is the meta-learner, an overarching model that helps in initializing and updating the meta-parameters, β^*. These parameters serve as a shared knowledge base that aids in swift adaptation across a myriad of tasks. In the e-commerce domain, where products constantly evolve and consumer preferences shift, the meta-learner aids in reducing the "cold-start" problem for new products, drawing insights from the plethora of data available across various other products.

Base Learners The base learners are models tailored to specific tasks, which in our context, might be predicting the demand for a new product launch or forecasting sales during a flash sale. Every task, be it associated with seasonal products or niche items, has unique characteristics. These base learners are initialized with the knowledge from the meta-learner and are then further trained on task-specific data, thereby combining global e-commerce insights with specific product nuances.

Ensemble and Fine-Tuning Fine-tuning is the art of refining the model parameters to enhance accuracy and precision. In our approach, post the meta and base learning stages, we adopt an ensemble technique coupled with fine-tuning. Here, the FiLM layer plays a pivotal role by leveraging the proxy data to modulate the learner parameters. This ensures that the models are not just generic but are tailored, accounting for intricate e-commerce dynamics. The ensemble of these fine-tuned models provides a robust forecast, reducing potential errors and capturing a wide array of patterns.

The strength of this method mainly lies in its ability to utilize shared structures across tasks while also adapting swiftly to unique task characteristics using the FiLM transformer and proxy data.

5. Experiment

In this section, we present a series of extensive experiments to assess the efficacy of our proposed F-FOMAML for Demand Prediction. We aim to address the following key research questions:

Research Question 1: How does the overall prediction performance of F-FOMAML compare to state-of-the-art baselines?

Research Question 2: To what extent do the various components we introduce, such as the selection method of proxy data (i.e., GNN versus clustering), impact the model's predictive capabilities?

By investigating these research questions, we provide a thorough evaluation of F-FOMAML and shed light on its performance in relation to existing approaches. Moreover, we analyze the contributions of individual components to showcase the influence of our proposed innovations on the overall predictive power of the model.

5.1. Experimental Setups

In the experimental settings, we discuss two real-world datasets and the method's performance assessment criteria. We employ the mean squared error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) [3, 8], to evaluate the performance.

Datasets. To validate our methodology, we sourced data from two major trading contexts, namely, the vending machine merchandising dataset and JD.com dataset from a renowned e-commerce platform (i.e., JD.com).

Vending Machine Merchandising Dataset. This dataset is derived from a private vending machine company. The dataset contains sales data from Mar 10, 2022, to April 20, 2022, for 246 products and 1715 vending machines. Each product from a specific vending machine has a base price (last for the first 20 days) and different adjusted prices (last for the last 20 days). The goal is to estimate the demand at the adjusted price given the demand at the base price.

Table 1. Experiment results of F-FOMAML using vending machine sales and JD.com data. Our F-FOMAML obtains the smallest error on the real-world dataset than the competing baselines.

Method	with GNN	MSE	MAE	MAPE
Linear Regression	No	0.6218	0.4782	0.2900
MLP	No	0.2811	0.2038	0.1499
MAML	No	0.2985	0.2143	0.1587
MeLU	No	0.2756	0.1934	0.1377
F-FOMAML (Ours)	No	**0.2345**	**0.1532**	**0.1206**
Linear Regression	No	0.8561	0.4235	**0.2033**
MLP	No	0.7164	0.4215	0.2047
MAML	No	5.5366	0.6088	0.4083
Time-series Regres.	No	6.0413	0.5963	0.3297
F-FOMAML (Ours)	No	**0.6089**	**0.3876**	0.2077

We use the category information (7 categories in total) for product features, and region (4 regions) and scene (8 scenes) information for vending machine feature. We use the last 10 days as the testing set, and the second to last 10 days as meta-training. This setting has no data leakage.

JD.com dataset. We work with the transactional records from JD.com, which offer a blend of both static and dynamic features related to the product (SKU) and order details for the month of March 2018 [34].

The *SKU table* contains information about the SKUs that were clicked at least once during March 2018. Each SKU entry has a unique SKU ID and is associated with a seller. For this study, 9,167 SKUs were selected. Each SKU possesses two pivotal attributes, which could, depending on the category, represent product features like SPF for face moisturizers or the number of personalized shaving modes for electric shavers.

The *Order table* encompasses details about unique customer orders within our designated product category from March 2018. This table elucidates specifics like order quantity, order date and time, SKU type, and the promised delivery time of the order. Additionally, it captures the product pricing and promotional activities, delineating the difference between the original and final unit price, thereby indicating the promotional discounts offered.

The goal is to predict the demand at the promotional price given the demand at the regular price. We use the category information (3 categories in total) for product features, and region (63 regions) information for location features. We use the last 15 days as testing, and the second to last 15 days as training.

Task Definition Considering the e-commerce platform, each task T, is depicted as: Product location or categorization (j), SKU ID or product identity (i), and Transactional details. The Transactional data further contains 1) Static features, that entail attributes of the SKU and characteristics of the order. 2) Dynamic features, that embrace the temporal sales data, including product prices and quantities

sold.

For our analysis, we split our tasks into the following partitions: 1) Training set $\mathcal{D}_{train}$, comprising data from regular sales days, this is used for initial model training. 2) Query set $\mathcal{D}_{query}$, which consists of slightly modified versions of tasks from $\mathcal{D}_{train}$ and facilitating the inner-loop adaptation. 3) Testing set $\mathcal{D}_{test}$, that incorporates data from peak sales periods and is earmarked for assessing model performance.

Baselines To assess the effectiveness of our proposed method, we compare it against various baselines. We categorize these baselines into different types: 1) Basic regression methods: Linear Regression model and time series regression (i.e., Autoregressive model [35]). 2) Neural network-based methods: we consider the Multi-Layer Perceptron network. 3) Transfer-learning based methods: We consider two current popular Transfer-learning based methods: MAML [12] and MeLU [20]. For MAML variants, We adapt the meta-learner using the base price-demand data as support. For MeLU [20], we build upon the MAML framework, MeLU can adapt to new products through a few local updates.

By considering these diverse baselines, we establish a comprehensive benchmark to facilitate a thorough evaluation of our proposed approach in comparison to state-of-the-art methods.

Implementation Details. In terms of GNN-based learning tasks, we aggregate order information for each product to generate the daily quantities sold during the one-month period of JD.com data. We also include the original and final price sequences as the time series features. Meanwhile, we also obtain each product's static features such as brand and attributes. Our forecasting task is set to predict the sales for future one day of each product using the past 16 days' information (i.e., $C = 16$ in Equation (6)).

The graph constructed uses products' brand information such that products under the same brand are connected. In this task, the graph contains 9159 nodes that represent all products having at least 1 unit of sale in the one-month period. We use two graph convolutional (GCN) layer to explore all 2-hop neighbors for each product, and together with its own static and dynamic features in the prediction task.

We train the graph-based model until convergence. The Adam optimizer with default settings is used for minimizing the Mean Absolute Error (MAE) loss with a batch size of 8 to 100 epochs. The final embeddings for each product is a numerical vector of length 90 which contains $50, 8, 32$ encoded features representing static, dynamic and graph-enhanced features respectively.

Model Evaluation and Training. With the meta-learning framework in place, we train the base learners on the proxy data and evaluate their performances using evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The meta-learner, which could be a neural network [14], support vector machine [7], or decision tree [4], selects the best base learners and their corresponding hyper-parameters based on the evaluation results.

Next, the selected base learners are fine-tuned on the available historical sales data from the target vending machine, if any, to adapt the model to the specific characteristics of the vending machine [11]. This fine-tuning step allows our model to better capture the unique relationships between features and sales in the target vending machine, leading to more accurate predictions and improved generalization to new tasks.

Table 2. Experiment results on real-world JD.com E-Commerce sales data with GNN features. MSE stands for Mean square error, MAE is Mean Absolute Error and MAPE is Mean absolute percentage error.

Method	with GNN	MSE	MAE	MAPE
Linear Regression	Yes	0.7633	0.4138	**0.1821**
MLP	Yes	0.7323	0.4199	0.1987
MAML	Yes	1.0752	0.4769	0.2345
Time-series Regres.	Yes	6.0413	0.5963	0.3297
F-FOMAML (Ours)	Yes	**0.6552**	**0.3713**	0.2117

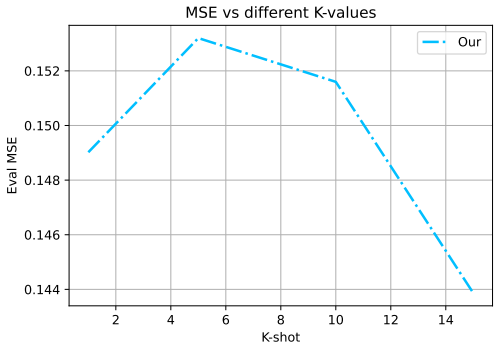

Figure 2. Evaluation of MSE vs. different values of k for the *k-shot* proxy data selection. The MSE goes up with an increase in the k values until it saturates, after which it starts reducing again as the value of k increases further.

5.2. Analysis of Results

This section delves into the empirical findings to address our research questions.

To address the first research question, we present the findings in Table 1. Here, both linear regression and Multi-Layer Perceptron (MLP) set a high standard as baseline models, rivaling the performance of the broader meta-learning framework, MAML. This observation might be rooted in MAML's tendency to overfit, especially under the limitations of a small feature set and limited historical data. In contrast, the MeLU approach [20] slightly outperforms

MAML [11]. This can be ascribed to MeLU's local update strategy, effectively mitigating overfitting risks. Noteworthy is the superior performance of our proposed method, which significantly surpasses existing benchmarks. This success highlights the critical importance of strategically utilizing proxy data to elevate prediction accuracy. Our results demonstrate that F-FOMAML advances beyond current forecasting methods, achieving a 26.24% reduction in prediction MAE on the vending machine dataset and 8.7% on the JD.com dataset, especially with domain-specific relation construction.

Notably, GNNs were not employed in the vending machine dataset analysis (Table 2) due to the dataset's lack of continuous time-series data, a prerequisite for GNNs to generate effective embeddings. Hence, a direct comparison with GNNs on the vending machine dataset is not presented.

Concerning the second research question, we probe the nuances of utilizing proxy data. This involves juxtaposing our results with those obtained using GNN-enhanced models, as detailed in Table 2. The strategic integration of GNNs for creating task-specific embeddings and forming inter-task connections based on these GNN-derived representations significantly enhances our understanding of task interrelations. This in-depth insight is instrumental in curating an optimized proxy dataset. In comparison to GNN benchmarks, our method registers an impressive 11.44% improvement in MAE. This underscores the impact of a comprehensive task understanding in refining the training effectiveness and predictive precision of F-FOMAML.

A comparison between Tables 1 and 2 reveals that our GNN-enhanced method closely matches, and even excels in MAE, against approaches anchored in domain knowledge, such as hierarchy structures. This observation solidifies our assertion that our GNN framework effectively mirrors, if not surpasses, correlations discerned through human domain expertise.

Ablation study. To better understand the effect of proxy data, we perform an ablation study for our method where we vary the k parameter for the *k-shot* proxy data selection and evaluate the Mean Squared Error for our method as the value of k changes. We show the results in Figure 2 and observe that the MSE value goes up with an initial increase in the k value, until it saturates, after which it starts decreasing again as the value of k increases, potentially indicating the merit of sampling larger magnitudes of the proxy data. Additional details regarding the training performance are presented in Figure 3, where our algorithm converges to a lower error than other baselines.

6. Conclusion

In this paper, we unveiled an innovative approach to tackle the intricate challenge of predicting demand during unique sales events, marked by distinct buying behaviors.

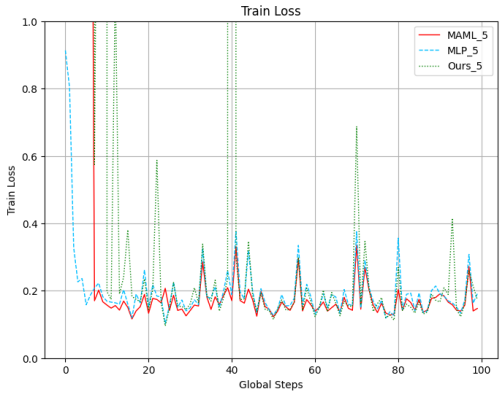

Figure 3. Training performance (MSE values) throughout training epochs over MAML, MLP and our proposed methods, where k is set to be 5.

The limited availability of event-specific historical data compounds this challenge, often making traditional sales data inadequate in portraying the specifics of such events. With this backdrop, we reconceived the problem of demand prediction within the framework of graph-augmented meta-learning. By leveraging the GNN-enhanced F-FOMAML algorithm, we managed to synergistically combine the robustness of meta-learning with the flexibility offered by the Feature-wise Linear Modulation (FiLM) layers. The result is a forecasting model that stands out, especially in scenarios defined by sparse historical data.

Empirical assessments not only attest to our model's edge over traditional forecasting methods but also highlight its versatile applicability, extending far beyond retail. The capability of our GNN-infused F-FOMAML suggests its potential in a range of applications, from bolstering online banking security mechanisms to amplifying the effectiveness of digital marketing campaigns.

In terms of empirical performance, F-FOMAML consistently outperforms prevailing forecasting techniques, marking an improvement in prediction MAE values by 26.24% on the vending machine dataset and 8.7% on the JD.com dataset, particularly when leveraging domain-specific for relation construction. Compared to GNN-based benchmarks, our approach demonstrates a commendable 11.44% enhancement in the MAE metric.

References

[1] Lennart Baardman, Igor Levin, Georgia Perakis, and Divya Singhvi. Leveraging comparables for new product sales forecasting. *Available at SSRN 3086237*, 2017. 2

[2] Jared Bishop, John Peters, and York Yannikos. Rapid response learning of humanitarian interventions. In *2014 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM)*, pages 9–15. IEEE, 2014. 2

[3] Alexei Botchkarev. A new typology design of performance metrics to measure errors in machine learning regression al-

gorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076, 2019. 6

[4] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. *CRC press*, 1984. 7

[5] Xinyu Cao and Juanjuan Zhang. Preference learning and demand forecast. *Marketing Science*, 40(1):62–79, 2021. 2

[6] Kevin Chang, Jingxuan Wu, Xinyang Yu, Ruichen Yu, Xiaoxin Chen, Weiwei Dai, and David Anastasiu. Chimera: Large-scale classification using machine learning, rules, and crowdsourcing. In *Proceedings of the VLDB Endowment*, volume 12, pages 193–206. VLDB Endowment, 2019. 2

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 7

[8] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016. 6

[9] Carson Eisenach, Yagna Patel, and Dhruv Madeka. Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention. *CoRR*, abs/2009.14799, 2020. 2

[10] Kris Johnson Ferreira, Bin Hong Alex Lee, and David Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & service operations management*, 18(1):69–88, 2016. 2

[11] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. 1, 2, 7, 8

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017. 7

[13] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073, 2012. 2

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. In *MIT Press*, 2016. 7

[15] Shivam Gupta, Kiran Ramesh, and Anil Kumar. Transfer learning for yield prediction. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 1109–1116. IEEE, 2016. 2

[16] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. In *Science*, volume 353, pages 790–794. American Association for the Advancement of Science, 2016. 2

[17] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697, 2018. 2

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net, 2017. 2

[19] Vincent Le Guen and Nicolas Thome. Probabilistic time series forecasting with structured shape and temporal diversity. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 1

[20] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *KDD*, pages 1073–1082. ACM, 2019. 7

[21] Jing Li, Yan Zhang, Zhenzhen Yang, and Shengjun Wang. Relation-aware meta-learning for e-commerce market segment demand prediction with limited records. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 2325–2334. ACM, 2020. 2

[22] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017. 2

[23] Bryan Lim, Sercan Ömer Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *CoRR*, abs/1912.09363, 2019. 2

[24] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105. PMLR, 2015. 2

[25] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and SY Philip. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013. 2

[26] Will Ma and David Simchi-Levi. Constructing demand curves from a single observation of bundle sales. In *International Conference on Web and Internet Economics*, pages 150–166. Springer, 2022. 2

[27] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. 1, 2

[28] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. 2

[29] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009. 2

[30] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, pages 3942–3951. AAAI Press, 2018. 5

[31] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019. 2

[32] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. 2

[33] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *ICLR*. OpenReview.net, 2021. 2

[34] Max Shen, Christopher S Tang, Di Wu, Rong Yuan, and Wei Zhou. Jd. com: Transaction-level data for the 2020 msom

data driven research challenge. *Manufacturing & Service Operations Management*, 2020. 6

[35] Robert H Shumway, David S Stoffer, and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000. 7

[36] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017. 2

[37] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning, 2017. 2

[38] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014. 2

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2962–2971, 2017. 2

[40] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016. 2

[41] Yaqing Wang, Qin Yao, and Ivor W Tsang. Meta-learning for online retail sales prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2073–2076. ACM, 2019. 2

[42] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017. 2

[43] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020. 2

[44] Sitan Yang, Malcolm Wolff, Shankar Ramasubramanian, Vincent Quenneville-Belair, Ronak Mehta, and Michael Mahoney. Geann: Scalable graph augmentations for multi-horizon time series forecasting. In *KDD 2023 Workshop on Deep Learning on Graphs*, 2023. 1, 2, 5

[45] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Learning to learn by remembering. In *Advances in Neural Information Processing Systems*, pages 1574–1584, 2019. 2