Enhancing E-commerce Supply Chain Management through Large Time Series Model

Shiyu Wang*†[1], Xinyue Zhong*[1], Jiawei Li*[2], Rongwei Liu[1], Yidong Feng[1], Congcong Hu[1]
Fan Huang[1], Zhou Ye [1]

[1]ByteDance
[2]The Hong Kong University of Science and Technology (Guangzhou)

Abstract

The rapid growth of the internet has driven increasing market demands and consumer expectations in E-commerce, making efficient supply chain management (SCM) essential for online businesses. To keep pace, accurate SCM forecasting has become crucial, helping companies anticipate demand, optimize inventory, and reduce costs. Traditional forecasting models often fail to address critical challenges in E-commerce SCM, such as data sparsity, long-tail distributions, and complex business scenarios. To bridge this gap, we introduce MoEChain, an innovative SCM framework powered by the large time series model Time-MoE. Designed to overcome critical obstacles, MoEChain integrates demand, supply, and logistics planning into a unified architecture, supporting forecasting tasks of any scale and length. Time-MoE leverages an mixture-of-experts (MoE) transformer with zero-shot inference and adaptation capabilities, enabling accurate predictions for demands with limited historical data and adjusting to diverse business scenarios. By pre-training on a meticulously curated balanced dataset, we effectively mitigate the impact of long-tail distributions. Extensive experiments on four real-world SCM datasets demonstrate that MoEChain achieves state-of-the-art forecasting performance, with its zero-shot predictions surpassing task-specific fully trained baselines. Further improvements of 13.06% in MSE are achieved through one-epoch fine-tuning. This work represents the first comprehensive application of a large time series model in SCM, showcasing its practical potential to support optimized decision-making.

CCS Concepts

• **Information systems** → **Temporal data**; • **Mathematics of computing** → **Time series analysis**; • **Computing methodologies** → **Neural networks**.

Keywords

Supply Chain Management, Large Time Series Model, Zero-Shot Forecasting

Correspondence to: S. Wang. **Email:** kwuking@gmail.com.

WWW'25 AI4TS Workshop, April 28-29, 2025, Sydney, NSW, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Shiyu Wang*†[1], Xinyue Zhong*[1], Jiawei Li*[2], Rongwei Liu[1], Yidong Feng[1], Congcong Hu[1] and Fan Huang[1], Zhou Ye [1]. 2025. Enhancing E-commerce Supply Chain Management through Large Time Series Model. In *Companion Proceedings of the ACM Web Conference Workshop 2025 (WWW'25 AI4TS Workshop), April 28-29, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The rapid growth of the internet has transformed modern commerce, fostering new business models and creating unprecedented connectivity across industries. E-commerce has been one of the most impacted sectors, becoming an essential part of everyday life. Web-based platforms like Taobao, JD.com, and ByteDance have revolutionized online shopping, particularly in regions such as China, driving rapid expansion in E-commerce. At the core of this success lies Supply Chain Management (SCM) [14], a vital function that orchestrates the flow of goods, information, and finances from raw material suppliers to consumers. To maintain a responsive and efficient supply chain, E-commerce companies must adapt swiftly to fluctuating demands, relying heavily on proactive planning. Accurate SCM forecasting [12, 16, 30] is essential, providing predictive insights into demand, inventory, and logistics that enable companies to anticipate needs and optimize resources. Leveraging advanced forecasting models, businesses can project future demand trends [34], optimize stock levels [4], and ensure product availability, while minimizing costs related to overstocking and stockouts [30]. These predictive capabilities enable seamless coordination between warehouses and fulfillment centers, enhancing supply chain agility and reducing operational costs through optimized inventory flow and transportation. Integrating forecasting models with web-based time series data, such as online demand patterns and inventory turnover, enables E-commerce businesses to make informed, data-driven decisions–a competitive necessity in dynamic online markets.

In SCM, decisions are inherently interconnected, with each stage influencing the next in a continuous loop. Previous research on SCM for web-based businesses can be broadly categorized into demand planning, supply planning, and integrated demand-supply planning. In demand planning, significant efforts have been made to predict sales or demand sequences for online platforms. Various time series models has been developed to capture temporal dynamics in sales data [2, 16, 21]. However, these methods often face challenges like cold-start scenarios [18, 34], especially when dealing with new products or markets where historical data is limited. To address these issues, transfer learning approaches [29, 30, 34] have been

proposed to enhance prediction accuracy by leveraging knowledge from related tasks or domains, thereby mitigating the effects of insufficient data. For supply planning, researchers have formulated inventory management as an optimization problem [4, 7, 20], aiming to maintain a balance between the supply and demand of Stock Keeping Units (SKUs) by optimizing replenishment decisions. These approaches typically rely on strong assumptions, such as customer demand following a fix distribution [33], which limit their applicability in the dynamic settings of web-based business. Accurate estimation of demand and inventory flow is critical for optimizing supply planning and maintaining operational efficiency. To better capture the interconnected nature of SCM decisions, some studies have explored integrated demand-supply planning. For instance, lin et al. [12] jointly predict purchase demand and supply within a unified framework, and similar methods have been applied in domains like e-scooter sharing and car-hailing services [6, 23].

Despite advancements in SCM forecasting, existing methods face critical challenges that limit their practical application in E-commerce environments:

- **Complex SCM scenarios:** SCM in E-commerce spans multiple tasks and scenarios, each with unique forecasting requirements. For instance, demand planning, supply planning, and logistics planning each involve different data structures and varying time horizons. Existing models often lack the flexibility to adapt across these scenarios without extensive retraining, hindering their scalability and operational efficiency in SCM applications.
- **Long-tail distribution:** A key characteristic of E-commerce is the long-tail distribution of demand, where a small number of popular items have abundant data while the majority of SKUs are infrequently purchased. Traditional forecasting models are typically optimized for well-represented items and struggle to accurately predict demand for long-tail SKUs, which are essential to manage for an effective supply chain strategy that covers the full product range.
- **Sparse data sources:** SCM for E-commerce requires data integration from numerous sources, including transaction logs, inventory levels, and logistics operations. While diverse, these data sources often suffer from sparsity, particularly for newer products with limited historical demands. The lack of sufficient and consistent data for these items complicates the training of reliable forecasting models, limiting their accuracy and applicability across all SKUs.

In this paper, we propose MoEChain, an innovative SCM framework equipped with a large time series model (LTM), Time-MoE [19], specifically designed to tackle the challenges of diverse SCM scenarios, long-tail distributions, and data sparsity. Our framework integrates demand planning, supply planning, and logistics planning within a unified architecture, providing a versatile, high-performance solution for E-commerce SCM that supports any-variate and any-length forecasting. As illustrated in Figure 1, the core of this architecture consists of an online business platform that consolidates heterogeneous data sources across the supply chain, including demand metrics, inventory flows, and logistics data, and an algorithm deployment platform that enables both online and offline inference of the proposed Time-MoE model.

To handle complex SCM scenarios, our solution integrates an mixture-of-experts (MoE) mechanism within a decoder-only transformer architecture, dynamically adapting to diverse forecasting tasks in an efficient and scalable manner. For the long-tail demand distribution, Time-MoE is pre-trained on a high-quality, well-curated enterprise proprietary dataset, incorporating careful preprocessing and sampling strategies to ensure a balanced domain distribution. Lastly, to address data sparsity, we leverage the zero-shot inference capability of Time-MoE and a lightweight adaptor for task-specific fine-tuning, enabling accurate predictions even for products with limited historical data. We validated the effectiveness of our framework using real-world E-commerce data from an E-commerce company. In three distinct business scenarios, our framework successfully performed zero-shot forecasting on four types of business time series, outperforming baselines that require training from scratch. Additionally, we introduced a lightweight adaptor tailored to specific business scenarios, further reducing forecasting error by 9.3% while maintaining inference efficiency. These results demonstrate the strong practical value and applicability of our framework in real-world E-commerce SCM. Our contributions are summarized as follows:

- We introduce MoEChain, an efficient and innovative supply chain management framework that integrates demand, supply, and logistics forecasting within a single architecture, powered by the large time series model Time-MoE.
- Leveraging the Time-MoE, MoEChain enables zero-shot, any-variate, and any-length forecasting, effectively addressing challenges such as data sparsity, long-tail distributions, and diverse operational scenarios. Additionally, we propose a lightweight adaptor to enhance adaptability.
- MoEChain achieves state-of-the-art zero-shot performance on four real-world datasets, surpassing fully trained baselines. Further improvements of 13.06% in MSE are achieved through one-epoch fine-tuning.

2 Related Work

2.1 General time series forecasting

Recent advancements in deep learning have expanded the range of time series forecasting models, leveraging various architectures to capture the temporal dependency. For instance, recurrent neural network (RNN)-based models [10, 35] and temporal convolution network (TCN)-based models [24, 28] have demonstrated strong performance in temporal pattern modeling. Multi-layer perceptron (MLP)-based models [11, 26, 31] offer simplicity and scalability, while attention-based methods [13, 15] dynamically focus on relevant features, achieving state-of-the-art results in diverse forecasting tasks. Despite their effectiveness, these models often require task-specific training and deployment, limiting their scalability in real-world settings characterized by data sparsity, long-tail distributions, and operational complexity. Recent efforts [1, 19, 27] have explored LTMs with zero-shot forecasting capabilities, showing promising adaptability without task-specific fine-tuning. However, their practical integration into industrial applications remains underexplored.

2.2 Time series forecasting in E-commerce

E-commerce time series forecasting plays a critical role in optimizing SCM, with key challenges such as data sparsity and long-tail distributions shaping the development of advanced methods. Existing forecasting work in E-commerce SCM has primarily focused on demand forecasting to anticipate customer needs. One line of research emphasizes transfer learning to address data scarcity by leveraging knowledge from auxiliary domains or tasks. For instance, RMLDP [18] employs meta-learning to transfer segment-specific knowledge for demand prediction in sparse market segments, while CATN [34] enhances cross-market forecasting by modeling cooperative and competitive interactions between markets. Another line of work focuses on incorporating external signals to improve prediction accuracy. Examples include embedding world events to mitigate forecasting errors [9] and leveraging time-sensitive features and user behavior data to dynamically refine predictions [8]. While these methods have advanced the field, they still rely on domain-specific configurations and struggle to scale across diverse applications, highlighting the need for frameworks that can seamlessly adapt to varied operational scenarios.

2.3 Supply chain management

SCM in E-commerce encompasses multiple interconnected components, with supply planning and logistics planning representing two critical areas of focus. Supply planning seeks to balance inventory replenishment with demand fulfillment under resource constraints. Recent methods, such as WIMS [33], optimize replenishment decisions for SKUs under resource constraints, while CD-PPO [4] employs decentralized reinforcement learning to efficiently manage shared inventory resources. However, these approaches often rely on fixed assumptions, such as predefined demand distributions or zero lead time [33], which restrict their adaptability to the dynamic and uncertain nature of E-commerce environments. Logistics planning, on the other hand, emphasizes optimizing delivery operations, particularly in dynamic scenarios such as last-mile logistics. For example, GCRL [25] utilizes cooperative reinforcement learning to enable efficient coordination of courier teams, while M^2G4RTP [3] leverages graph-based architectures to jointly predict delivery routes and times. In contrast to the aforementioned studies, we directly predicts critical variables in supply planning and logistics planning, providing actionable insights that enhance decision-making efficiency in E-commerce SCM.

3 Methodology

In this paper, we propose MoEChain, an innovative SCM framework to meet the multifaceted forecasting demands of SCM, including demand planning, supply planning, and logistics planning. As shown in Figure 1, the core of MoEChain consists of an online business platform that consolidates heterogeneous data sources across the supply chain and an algorithm deployment platform that enables both online and offline inference of the proposed Time-MoE model. This framework empowers robust, data-driven decision-making, enhancing supply chain efficiency and resilience, ensuring that each component, from inventory flow to transportation logistics, is aligned with real-time and future demands.

3.1 Preliminaries

We introduce the key concepts that form the foundation of our framework. This includes defining the core entities and systems involved in E-commerce SCM, along with their roles and interactions:

- **Warehouse**: Serving as the central hub of the E-commerce supply chain, the warehouse manages the storage, organization, and distribution of inventory. It coordinates the receipt of inbound goods from suppliers and oversees the dispatch of outbound orders to fulfillment centers.
- **Fulfillment Center**: The fulfillment center receives shipments from the warehouse via trucks, handling the picking, packing, and preparation of products for final shipment to customers or other distribution points.
- **Stock Keeping Unit (SKU)**: An SKU is a unique identifier assigned to each distinct product variant available for sale.
- **E-commerce Transaction System (ETS)**: This subsystem records transaction-level sales data for individual SKUs, which are instrumental for demand planning. This data includes dynamic sales patterns for a range of SKUs and serves as a foundation for predicting future demand.
- **Warehouse Management System (WMS)**: The WMS monitors inventory flow, capturing both inbound and outbound data for multiple warehouses. This data is essential for optimizing stock levels, managing supply, and ensuring timely order fulfillment.
- **Logistics Management System (LMS)**: The LMS records transfer volumes across different transportation routes, specifically tracking inter-warehouse transfers and movements from warehouses to fulfillment centers. This data facilitates route optimization and efficient resource allocation within logistics planning.

These entities and systems generate diverse time series data, encompassing sales transactions, inventory movements, and transportation records. These real-time and historical data streams are processed as multi-variate time series $\mathbf{X} \in \mathbb{R}^{M \times T}$, where each data type contributes unique insights into supply chain dynamics:

- **Sales data**: Captures daily transactional records for M SKUs, often numbering in the hundreds. This data reflects complex demand patterns shaped by seasonality, promotions, and shifting consumer behavior.
- **Inbound data**: Tracks goods entering warehouses, with M representing the number of inbound categories (e.g., apparel, non-apparel) or warehouse zones being monitored. Inbound data exhibits cyclical trends driven by supplier schedules and replenishment cycles, with occasional spikes.
- **Outbound data**: Refers to the quantity of goods picked, packed, and shipped from warehouses in response to customer orders, with M denoting the number of destinations served. Temporal variations arise from fluctuations in order volumes and supply chain constraints.
- **Transportation data**: Represents the transfer of goods from warehouses to fulfillment centers, with M indicating transportation routes. This data shows periodic trends but can fluctuate due to demand shifts or route disruptions.

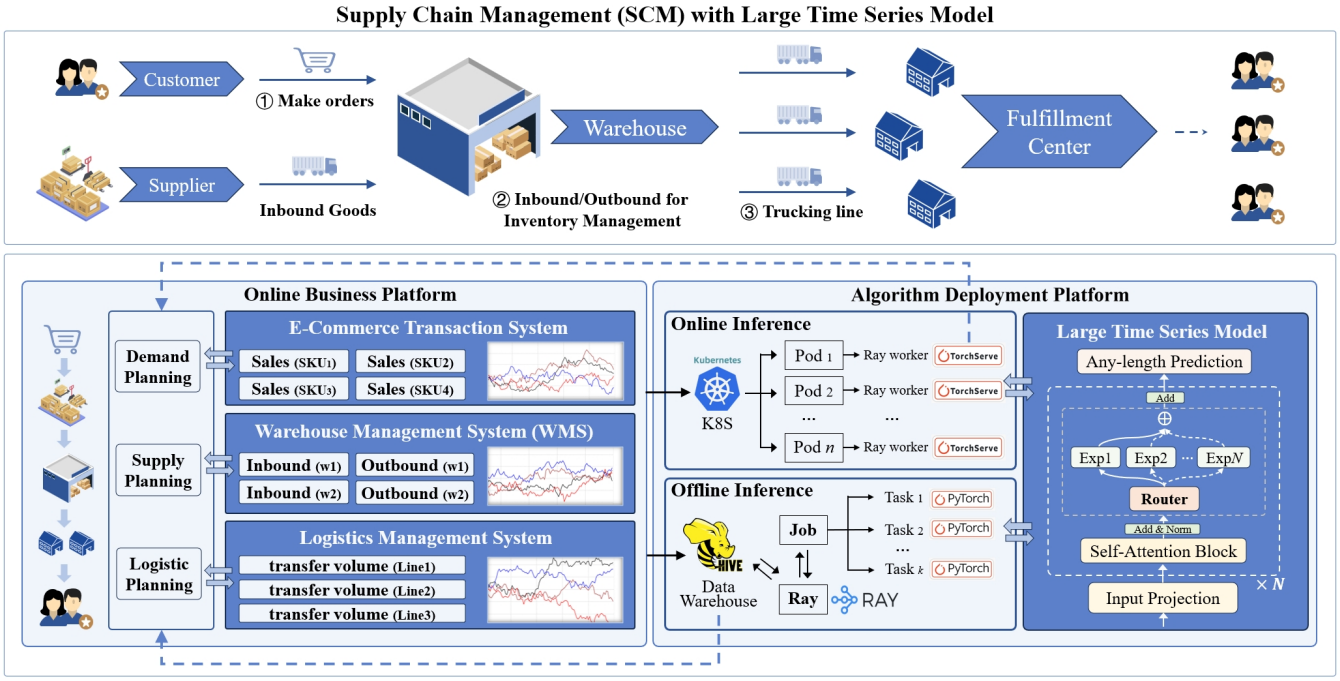

Figure 1: Overview of the MoEChain. The framework supports decision-making in E-commerce SCM, including demand, supply, and logistics planning, through an online business platform and algorithm deployment platform. A LTM is deployed to enable efficient online and offline inference for any-length predictions.

3.2 Framework Overview

As shown in Figure 1, MoEChain integrates the online business platform and the algorithm deployment platform to provide scalable and efficient solutions for SCM in E-commerce. The online business platform is composed of three core subsystems: ETS, WMS, and LMS. These subsystems generate and manage diverse time series data, including sales patterns, inventory movements, and transportation volumes, which are critical for supporting demand planning, supply planning, and logistics planning.

To empower advanced decision-making, the algorithm deployment platform utilizes the zero-shot predictive capabilities of the Time-MoE model. Built on a decoder-only mixture-of-experts (MoE) architecture, Time-MoE delivers computationally efficient and accurate forecasts across diverse time series tasks. Its zero-shot inference capability enables predictions without task-specific fine-tuning, offering flexibility for any-variate and any-length forecasting. This adaptability makes Time-MoE well-suited to real-world E-commerce challenges, such as handling sparse datasets, addressing long-tail distributions, and adapting to complex and varied operational requirements.

The algorithm deployment platform supports two modes of operation: online inference, which delivers real-time predictions to integrate seamlessly with business operations, and offline inference, which ensures stable and reliable forecasts for batch processing. Further implementation details are provided in Section 3.5.

3.3 Time-MoE Architecture

Previous studies [1, 5, 27] utilizing transformer-based architectures face substantial computational overhead during inference due to the activation of all model parameters. In online E-commerce systems, inference speed and prediction accuracy are both critical, as they directly influence real-time decision-making and operational efficiency. To address these challenges, we propose a decoder-only transformer architecture integrated with an MoE mechanism. Our model comprises three core components: an input projection layer, decoder blocks, and an any-length prediction module. The input projection layer maps heterogeneous time series data into a shared embedding space, enabling flexible handling of multi-variate and varying-length sequences. The decoder blocks employ an MoE mechanism that selectively activates a subset of expert networks tailored to each business scenario, significantly reducing computational load while preserving model capacity. Finally, the any-length prediction module enables flexible forecasting horizons, supporting both short- and long-term predictions essential for dynamic E-commerce operations.

3.3.1 ***Input embedding.*** Previous works commonly employ patching techniques to generate patch-wise embeddings for time series data, capturing local semantics and reducing computational complexity. However, these methods generally lack adaptability to inputs of varying lengths and often require predefined operations, such as padding [5], quantization [1], or multi-patch size strategies [27], which limits their flexibility in handling time series data of

arbitrary lengths in E-commerce applications. To address these limitations, we propose an embedding approach that directly maps each time point into a unified embedding space, eliminating the need for length-specific preprocessing. Formally, the input embedding is defined as:

$$\mathbf{H}^0 = \text{Swish}\left(\mathbf{x}\mathbf{W}^\top\right) \otimes \left(\mathbf{x}\mathbf{V}^\top\right), \tag{1}$$

where $\mathbf{H}^0 \in \mathbb{R}^{T\times D}$ is the resulting embedding matrix, $\mathbf{x} \in \mathbb{R}^T$ represents a univariate time series of length T, Swish [17] is the activation function, and $\mathbf{W}, \mathbf{V} \in \mathbb{R}^{D\times 1}$ are learnable weight matrices. This formulation enables each time point to be independently mapped to the latent space, enhancing adaptability to input sequences of any length.

3.3.2 ***Decoder Blocks.*** Our model comprises a stack of L *Blocks* connected via residual connections, where each *Block* is a modified version of the standard transformer block. Specifically, we employ RMSNorm [32] instead of the LayerNorm to enhance training stability. Additionally, we utilize rotary positional embeddings [22] in place of absolute positional encodings to improve the model's flexibility in sequence length. The feed-forward network (FFN) in each block is replaced with an MoE layer. Then, the forward propagation of the l-th *Blocks* can be formalized as:

$$\begin{aligned}\mathbf{H}^l &= \text{Block}(\mathbf{H}^{l-1}) + \mathbf{H}^{l-1} \\ &= \text{SelfAttn}\left(\text{Norm}\left(\mathbf{H}^{l-1}\right)\right) + \text{MoE}\left(\text{Norm}\left(\mathbf{H}^{l-1}\right)\right) + \mathbf{H}^{l-1},\end{aligned} \tag{2}$$

where SelfAttn denotes the self-attention layer, and Norm represents RMSNorm. To efficiently handle diverse patterns in E-commerce time series data while maintaining computational efficiency, we incorporate an MoE mechanism into our decoder-only transformer architecture.

The MoE consists of $N+1$ expert networks, denoted as $\{\text{Expert}_1, \text{Expert}_2, \ldots, \text{Expert}_{N+1}\}$, where each Expert is implemented as a standard FFN. The first N experts are sparsely activated based on the input at each time step, and the $(N+1)$-th expert acts as a shared expert that does not participate in the routing mechanism and processes all time steps.

Routing Mechanism. We first compute routing logits to determine which experts to activate for each time step. This is achieved by projecting the input through a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{D\times N}$:

$$\mathbf{R} = \hat{\mathbf{H}}^{l-1}\mathbf{W}, \quad \mathbf{R} \in \mathbb{R}^{T\times N}, \quad \hat{\mathbf{H}}^{l-1} = \text{Norm}(\mathbf{H}^{l-1}). \tag{3}$$

Next, we apply the softmax function along the expert dimension to obtain the routing probabilities:

$$\mathbf{P} = \text{Softmax}(\mathbf{R}), \quad \mathbf{P} \in \mathbb{R}^{T\times N}. \tag{4}$$

For each time step t, we select the top k experts with the highest routing probabilities $\mathbf{P}_{t,:}$, resulting in a binary mask $\mathbf{M} \in \{0,1\}^{T\times N}$:

$$\mathbf{M}_{t,i} = \begin{cases} 1, & \text{if } i \in \text{TopK}\left(\mathbf{P}_{t,:}, k\right), \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Expert Computation and Aggregation. Each activated expert Expert_i $(i = 1, \ldots, N)$ processes its assigned time steps. Let $\mathcal{T}_i = \{t \mid \mathbf{M}_{t,i} = 1\}$ be the set of time steps assigned to expert i. The expert outputs are computed as:

$$\mathbf{O}_i = \text{Expert}_i\left(\hat{\mathbf{H}}^{l-1}_{\mathcal{T}_i}\right), \quad \mathbf{O}_i \in \mathbb{R}^{|\mathcal{T}_i|\times D}, \tag{6}$$

where $\hat{\mathbf{H}}^{l-1}_{\mathcal{T}_i}$ denotes the inputs corresponding to $\mathcal{T}_i$. The shared expert Expert_{N+1} processes all time steps:

$$\mathbf{O}_{N+1} = \text{Expert}_{N+1}\left(\hat{\mathbf{H}}^{l-1}\right), \quad \mathbf{O}_{N+1} \in \mathbb{R}^{T\times D}. \tag{7}$$

The final output of the MoE layer aggregates the contributions from the activated experts and the shared expert. For each time step t, the output is computed as:

$$\begin{aligned}\mathbf{H}^l_t &= \text{MoE}(\hat{\mathbf{H}}^l)_t = \alpha_t \cdot \mathbf{O}_{N+1,t} + \sum_{i=1}^{N} \mathbf{M}_{t,i} \cdot \mathbf{P}_{t,i} \cdot \mathbf{O}_{i,t}, \\ \alpha_t &= \text{Sigmoid}(\hat{\mathbf{H}}^{l-1}_t \mathbf{W}), \quad \mathbf{W} \in \mathbb{R}^{D\times 1},\end{aligned} \tag{8}$$

where $\mathbf{O}_{i,t}$ represents the output of expert i at time step t (if $t \in \mathcal{T}_i$) and zero otherwise. This formulation ensures that each time step is influenced only by its assigned experts.

3.3.3 ***Any-length Prediction.*** In E-commerce operations, forecasting over flexible horizons is essential for effective supply chain and logistics management. Different business strategies require predictions at various time scales; for example, short-term forecasts are vital for inventory replenishment, while long-term projections inform strategic planning and market expansion. To accommodate these requirements, we propose an auto-regressive prediction framework employing j specialized prediction heads, denoted as $\{\text{Head}_1, \ldots, \text{Head}_j\}$. Each prediction head Head_i is tailored for a fixed horizon length h_i, responsible for forecasting the next h_i time steps. During training, we jointly optimize all heads by minimizing the average forecasting error. During inference, to achieve a desired forecasting horizon H, we generate predictions auto-regressively by sequentially selecting the largest possible h_i that does not exceed the remaining horizon. The forecast is constructed as:

$$\hat{\mathbf{x}}_{T+1:T+H} = \text{Concat}\left(\text{Head}_{i_1}\left(\mathbf{H}^L_{1:T}\right), \text{Head}_{i_2}\left(\mathbf{H}_{1:T+h_{i_1}}\right), \ldots\right), \tag{9}$$

where each Head is selected greedily based on the remaining horizon, and Concat denotes the concatenation of the predicted sequences. Empirically, each Head is implemented as a single-layer FFN. By auto-regressively incorporating the predicted values back into the model input, our framework effectively captures dependencies across multiple time scales.

3.4 Time-MoE Adaptor

In E-commerce SCM, future covariates are also crucial for enhancing forecasting accuracy. Examples include time-related features such as the month of the year, the day of the week, and forward-looking procurement plans from operations teams. To enhance Time-MoE's predictions, we propose an optional lightweight adaptor that incorporates future covariates. Given the initial forecast $\hat{\mathbf{x}}_{T+1:T+H}$ generated auto-regressively by Time-MoE, where

$\hat{\mathbf{x}}_{T+1:T+H} \in \mathbb{R}^{M\times H}$, and m covariates $\{\mathbf{z}_1, \cdots, \mathbf{z}_m\}$, where $\mathbf{z}_i \in \mathbb{R}^H$, the adaptor refines the prediction as follows:

$$\hat{\mathbf{x}}_{T+1:T+H} = \text{Adaptor}([\hat{\mathbf{x}}_{T+1:T+H}; \mathbf{z}_1; \cdots; \mathbf{z}_m]), \tag{10}$$

where $[\cdot;\cdot]$ denotes concatenation along the variate dimension, and Adaptor is a two-layer FFN with ReLU activation, mapping inputs from $\mathbb{R}^{(M+m)\times H}$ to $\mathbb{R}^{M\times H}$. By leveraging covariate inputs, the adaptor enables efficient test-time fine-tuning, enhancing forecasting accuracy in dynamic SCM environments.

3.5 Model Deployment

3.5.1 ***Pre-training.*** We develope a high-quality time series dataset from enterprise proprietary data sources, to train foundation models for diverse forecasting tasks. We address data imbalance and long-tail distribution by applying down-sampling to overrepresented datasets, ensuring a balanced representation across domains and observation values during training. This strategy allows for proportionate sampling of batches, mitigating biases and promoting more robust model training. For deployment, we use Time-MoE$_{base}$ (50 million parameters) and Time-MoE$_{large}$ (200 million parameters).

3.5.2 ***Deployment.*** The deployment platform is designed to support ETS, WMS, and LMS with advanced decision support by leveraging the zero-shot predictive capabilities of our pre-trained Time-MoE. This platform operates in two distinct modes, **online** and **offline**, each tailored to specific operational requirements to maximize system responsiveness and computational efficiency:

- **Online Inference**: The online inference mode is structured for real-time forecasting to support time-sensitive decision-making in scenarios with fluctuating demand. The system employs Kubernetes (K8S)[1] for managing containerized services, dynamically scaling the number of pods according to workload requirements. Each pod is connected to a Ray worker[2], which orchestrates the distribution of inference tasks to ensure efficient utilization of resources. Within each pod, TorchServe[3] manages model serving, allowing the Time-MoE to deliver predictions with low latency.
- **Offline Inference**: The offline inference mode is designed for large-scale, compute-intensive forecasting tasks suited for long-term planning and periodic analysis. Integrated with Hive[4], the system uses Ray to manage batch jobs, distributing tasks across multiple nodes to enable parallel processing. Offline inference relies on PyTorch to execute batch processing, with Ray workers assigned to each task to optimize task distribution and maintain fault tolerance.

4 Experiment

4.1 Experimental Settings

4.1.1 ***Datasets.*** We evaluate MoEChain on real-world SCM data from an E-commerce company, covering three key business scenarios: demand planning, supply planning, and logistics planning. The datasets span from January 1, 2020, to December 31, 2023, featuring diverse time series characteristics to ensure a comprehensive assessment of SCM tasks. For demand planning, we use SKU-level sales time series with 935 variables per sample. For supply planning, we analyze inbound and outbound inventory flows from the warehouse, comprising 3 variables for inbound flows and 16 for outbound flows, capturing inventory movement dynamics. For logistics planning, we examine transportation time series (i.e., trucking lines) with 8 variables per sample. The datasets are split into training, validation, and test sets following a 7:1:2 ratio.

4.1.2 ***Metrics.*** We assess performance using three standard metrics: Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values, penalizing larger errors more heavily; Mean Absolute Error (MAE), which evaluates the average magnitude of errors without considering their direction for an intuitive measure of accuracy; and Mean Absolute Percentage Error (MAPE), which provides a relative accuracy measure by calculating percentage errors, making it suitable for datasets of varying scales.

4.1.3 ***Baselines.*** We compare Time-MoE against five state-of-the-art forecasting models. The baselines include the MLP-based model DLinear [31] and TimeMixer [26], the CNN-based model TimesNet [28], and the attention-based models iTransformer [13] and PatchTST [15].

4.2 Zero-Shot Forecasting

As shown in Table 1, we test the forecasting performance of Time-MoE$_{large}$ and baselines on four SCM time series datasets.

4.2.1 ***Implementation Details.*** For Time-MoE$_{large}$, the prediction heads, number of experts , hidden dimension D, number of decoder blocks L, and top-k expert selection k are consistently set to $\{1, 8, 32, 64\}$, 8, 768, 12, and 2, respectively. Baselines are trained using the Adam optimizer with a learning rate of 0.001. For Time-MoE$_{large}$, zero-shot forecasting is performed directly on the test set without fine-tuning, leveraging a Nvidia® A100 GPU for inference.

4.2.2 ***Main Results.*** The results in Table 1 highlight the superior performance of Time-MoE$_{large}$ across all tasks and forecasting horizons of one week, two weeks, and four weeks. On average, Time-MoE$_{large}$ achieves the lowest MAE of 0.613 and MSE of 1.240, outperforming the best baseline, iTransformer, by **9.7%** in MAE and **11.9%** in MSE. For the one-week horizon, Time-MoE$_{large}$ achieves an average MAE of 0.539 and MSE of 1.801, surpassing iTransformer by **7.6%** in MAE and **10.5%** in MSE. Similarly, for the four-week horizon, it achieves an average MAE of 0.673 and MSE of 2.662, outperforming iTransformer by **9.1%** in MAE and **12.3%** in MSE. These results underscore the robustness of Time-MoE$_{large}$ across varying timescales and resolutions. While certain baselines like PatchTST show strengths on specific datasets, such as excelling in the Outbound dataset, Time-MoE$_{large}$ demonstrates consistent generalization across all datasets and tasks. This universal adaptability eliminates the need for deploying and training separate models for different scenarios, significantly simplifying model deployment and maintenance. By effectively addressing challenges such as data

[1] https://kubernetes.io/
[2] https://www.ray.io/
[3] https://pytorch.org/serve/
[4] https://hive.apache.org/

Table 1: Time series forecasting tasks. The forecasting horizons are: {7, 14, 28}. The context length is 96. The models on the left side of the vertical lines represent zero-shot approaches, while those on the right are fully retrained models. The best performance is highlighted in red, and the second-best is underlined.

Methods		Time-MoE		TimeMixer		iTransformer		PatchTST		DLinear		TimesNet	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
SKU	7	**0.247**	**0.238**	0.252	0.251	0.253	0.247	0.278	0.262	0.381	0.413	0.373	0.461
	14	0.280	0.304	0.277	0.308	**0.269**	**0.297**	0.292	0.306	0.379	0.445	0.387	0.501
	28	0.331	0.398	**0.286**	**0.368**	0.302	0.392	0.324	0.390	0.402	0.516	0.414	0.583
Inbound	7	**0.538**	**0.440**	0.619	0.542	0.639	0.569	0.600	0.539	0.678	0.725	0.667	0.646
	14	**0.614**	**0.576**	0.704	0.746	0.742	0.809	0.744	0.815	0.687	0.738	0.732	0.831
	28	0.671	0.660	**0.634**	**0.609**	0.728	0.758	0.722	0.743	0.660	0.668	0.684	0.703
Outbound	7	**0.256**	**0.159**	0.339	0.275	0.312	0.204	0.292	0.178	0.347	0.250	0.407	0.329
	14	**0.296**	0.214	0.386	0.343	0.314	0.225	0.297	**0.195**	0.370	0.289	0.376	0.301
	28	0.395	0.331	0.403	0.373	0.382	0.325	**0.342**	**0.261**	0.405	0.352	0.381	0.300
Transportation	7	**1.117**	**3.366**	1.616	5.404	1.237	3.631	1.290	3.753	1.309	3.816	1.308	3.942
	14	**1.222**	**3.540**	1.735	5.471	1.338	3.837	1.411	4.141	1.368	3.906	1.380	3.947
	28	1.391	4.660	**1.387**	**4.332**	1.629	5.589	1.606	5.326	1.589	5.376	1.701	5.841
Average		**0.613**	**1.240**	0.720	1.585	0.679	1.407	0.683	1.409	0.715	1.458	0.734	1.532

sparsity, long-tail SKU distributions, and diverse business scenarios, Time-MoE$_{large}$ establishes itself as a scalable, efficient, and versatile solution for forecasting problems in SCM.

4.3 Fine-tuning Performance

Table 2: Performance comparison between zero-shot Time-MoE$_{large}$ and full fine-tuned Time-MoE$_{base}$. The best performance for each row is highlighted in bold.

Dataset		Time-MoE$_{large}$		Time-MoE$_{base}$ †		*Improv.* (%)	
		MAE	MSE	MAE	MSE	MAE	MSE
SKU	7	0.247	0.268	**0.241**	**0.250**	2.43% ↑	7.2% ↑
	14	0.280	0.304	**0.259**	**0.300**	7.50% ↑	1.32% ↑
	28	0.331	0.398	**0.287**	**0.392**	13.29% ↑	1.51% ↑
Inb.	7	0.538	0.440	**0.498**	**0.385**	7.43% ↑	12.50% ↑
	14	0.614	0.576	**0.572**	**0.504**	6.84% ↑	12.50% ↑
	28	0.671	0.660	**0.632**	**0.601**	5.82% ↑	8.94% ↑
Outb.	7	0.256	0.159	**0.249**	**0.148**	2.73% ↑	6.92% ↑
	14	0.296	0.214	**0.288**	**0.205**	2.70% ↑	4.21% ↑
	28	0.395	0.331	**0.365**	**0.297**	7.59% ↑	10.27% ↑
Transp.	7	1.117	3.366	**0.990**	**2.684**	11.37% ↑	20.27% ↑
	14	1.222	3.540	**1.033**	**2.878**	15.45% ↑	18.69% ↑
	28	1.391	4.660	**1.196**	**3.900**	14.03% ↑	16.34% ↑
Average	-	0.613	1.240	**0.580**	**1.078**	5.39% ↑	13.06% ↑

† represents the full fine-tuned Time-MoE$_{base}$, which was fine-tuned for only one epoch across all datasets.

4.3.1 ***Implementation Details.*** The fine-tuning strategy for Time-MoE$_{base}$ differs from conventional approaches to better align with SCM business requirements while preserving the model's generalization capabilities. Specifically, Time-MoE$_{base}$ was fully fine-tuned for one epoch on the shuffled training sets of all datasets. This strategy ensures its adaptability to various SCM scenarios without overfitting to specific tasks. Time-MoE$_{large}$ operates in a zero-shot setting, directly evaluated on the test set without any task-specific training.

4.3.2 ***Results.*** The results in Table 2 illustrate the performance gains achieved through fine-tuning the Time-MoE$_{base}$ model compared to its zero-shot larger variant Time-MoE$_{large}$. Fine-tuning results in consistent improvements across all datasets and forecasting horizons. For SKU-level forecasting, fine-tuning reduces MAE by **7.5%** on the 28-day horizon, improving from 0.331 to 0.287. Similarly, in the Inbound dataset, fine-tuning lowers MSE by **8.9%** on the 7-day horizon, decreasing from 0.440 to 0.385. The Transportation dataset shows the most significant improvement, where fine-tuning reduces MAE by **10.9%** and MSE by **16.3%** on the 7-day horizon. On average, fine-tuning improves MAE and MSE by **5.4%** and **13.1%**, respectively, demonstrating its effectiveness in enhancing prediction accuracy while leveraging the pre-trained model's zero-shot capabilities. These results underline the scalability and adaptability of Time-MoE for real-world applications.

5 Conclusion

In this work, we introduced MoEChain, a comprehensive SCM framework powered by the large time series model, Time-MoE. MoEChain addresses key challenges in E-commerce SCM, including data sparsity, long-tail distributions, and diverse business scenarios, by unifying demand, supply, and logistics planning in a single architecture. With an MoE transformer, Time-MoE delivers state-of-the-art zero-shot, any-variate, and any-length forecasting. Pre-trained on a curated, balanced dataset, Time-MoE effectively handles long-tail distributions while ensuring robust performance. Extensive experiments on four real-world SCM datasets show that MoEChain achieves state-of-the-art performance, with zero-shot predictions surpassing fully trained baselines and a 13.06% MSE reduction through one-epoch fine-tuning. MoEChain represents a pioneering application of large time series models in SCM, offering efficient and accurate forecasting for modern E-commerce, empowering data-driven decision-making in real-world operations.

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. 2024. Chronos: Learning the language of time series. arXiv preprint arXiv:2403.07815 (2024).

[2] Kasun Bandara, Peibei Shi, Christoph Bergmeir, Hansika Hewamalage, Quoc Tran, and Brian Seaman. 2019. Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III 26. Springer, 462–474.

[3] Tianyue Cai, Huaiyu Wan, Fan Wu, Haomin Wen, Shengnan Guo, Lixia Wu, Haoyuan Hu, and Youfang Lin. 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). IEEE, 3296–3308.

[4] Yuandong Ding, Mingxiao Feng, Guozi Liu, Wei Jiang, Chuheng Zhang, Li Zhao, Lei Song, Houqiang Li, Yan Jin, and Jiang Bian. 2022. Multi-agent reinforcement learning with shared resources for inventory management. arXiv preprint arXiv:2212.07684 (2022).

[5] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. 2024. Moment: A family of open time-series foundation models. arXiv preprint arXiv:2402.03885 (2024).

[6] Suining He and Kang G Shin. 2022. Socially-equitable interactive graph information fusion-based prediction for urban Dockless E-scooter sharing. In Proceedings of the ACM Web Conference 2022. 3269–3279.

[7] Massil Hihat, Stéphane Gaïffas, Guillaume Garrigos, and Simon Bussy. 2024. Online inventory problems: beyond the iid setting with online convex optimization. Advances in Neural Information Processing Systems 36 (2024).

[8] Jiarui Jin, Xianyu Chen, Weinan Zhang, Junjie Huang, Ziming Feng, and Yong Yu. 2022. Learn over past, evolve for future: Search-based time-aware recommendation with sequential behavior data. In Proceedings of the ACM Web Conference 2022. 2451–2461.

[9] Dan Kalifa, Uriel Singer, Ido Guy, Guy D Rosin, and Kira Radinsky. 2022. Leveraging world events to predict e-commerce consumer demand under anomaly. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. 430–438.

[10] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In The 41st international ACM SIGIR conference on research & development in information retrieval. 95–104.

[11] Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. 2023. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. arXiv preprint arXiv:2302.04501 (2023).

[12] Li Lin, Zhiqiang Lu, Shuai Wang, Yunhuai Liu, Zhiqing Hong, Haotian Wang, and Shuai Wang. 2024. MulSTE: A Multi-view Spatio-temporal Learning Framework with Heterogeneous Event Fusion for Demand-supply Prediction. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 1781–1792.

[13] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023).

[14] John T Mentzer, William DeWitt, James S Keebler, Soonhong Min, Nancy W Nix, Carlo D Smith, and Zach G Zacharia. 2001. Defining supply chain management. Journal of Business logistics 22, 2 (2001), 1–25.

[15] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In International Conference on Learning Representations.

[16] Yan Qi, Chenliang Li, Han Deng, Min Cai, Yunwei Qi, and Yuming Deng. 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.

[17] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017).

[18] Jiatu Shi, Huaxiu Yao, Xian Wu, Tong Li, Zedong Lin, Tengfei Wang, and Binqiang Zhao. 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.

[19] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. 2024. Time-MoE: Billion-Scale Time Series Foundation Models with Mixture of Experts. arXiv preprint arXiv:2409.16040 (2024).

[20] Kazuma Shimizu, Junya Honda, Shinji Ito, and Shinji Nakadai. 2024. Learning with Posterior Sampling for Revenue Management under Time-varying Demand. arXiv preprint arXiv:2405.04910 (2024).

[21] Balpreet Singh, Pawan Kumar, Nonita Sharma, and KP Sharma. 2020. Sales forecast for amazon sales with time series modeling. In 2020 first international conference on power, control and computing technologies (ICPC2T). IEEE, 38–43.

[22] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing 568 (2024), 127063.

[23] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. 2017. DeepSD: Supply-demand prediction for online car-hailing services using deep neural networks. In 2017 IEEE 33rd international conference on data engineering (ICDE). IEEE, 243–254.

[24] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In The eleventh international conference on learning representations.

[25] Hai Wang, Shuai Wang, Yu Yang, and Desheng Zhang. 2023. Gcrl: Efficient delivery area assignment for last-mile logistics with group-based cooperative reinforcement learning. In 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 3522–3534.

[26] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. arXiv preprint arXiv:2405.14616 (2024).

[27] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified training of universal time series forecasting transformers. arXiv preprint arXiv:2402.02592 (2024).

[28] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186 (2022).

[29] Tongwen Wu, Yu Yang, Yanzhi Li, Huiqiang Mao, Liming Li, Xiaoqing Wang, and Yuming Deng. 2021. Representation Learning for Predicting Customer Orders. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 3735–3744.

[30] Borui Ye, Shuo Yang, Binbin Hu, Zhiqiang Zhang, Youqiang He, Kai Huang, Jun Zhou, and Yanming Fang. 2022. Gaia: Graph neural network with temporal shift aware attention for gross merchandise value forecast in e-commerce. In 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 3320–3326.

[31] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In Proceedings of the AAAI conference on artificial intelligence, Vol. 37. 11121–11128.

[32] Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. Advances in Neural Information Processing Systems 32 (2019).

[33] Chuheng Zhang, Xiangsen Wang, Wei Jiang, Xianliang Yang, Siwei Wang, Lei Song, and Jiang Bian. [n. d.]. Whittle Index with Multiple Actions and State Constraint for Inventory Management. In The Twelfth International Conference on Learning Representations.

[34] Lei Zhang, Wang Xiang, Chuang Zhao, Hongke Zhao, Rui Li, and Runze Wu. 2022. Co-promotion Predictions of Financing Market and Sales Market: A Cooperative-Competitive Attention Approach. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 9040–9047.

[35] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. 2017. LSTM network: a deep learning approach for short-term traffic forecast. IET intelligent transport systems 11, 2 (2017), 68–75.