TAT: Temporal-Aligned Transformer for Multi-Horizon Peak Demand Forecasting

Zhiyuan Zhao*
leozhao1997@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Sitan Yang*
syang@keystonestrategy.com
Keystone AI
New York, NY, USA

Kin G. Olivares
kigutie@amazon.com
Amazon
New York, NY, USA

Boris N. Oreshkin
oreshkin@amazon.com
Amazon
New York, NY, USA

Stan Vitebsky
vitebsky@amazon.com
Amazon
New York, NY, USA

Michael W. Mahoney
zmahmich@amazon.com
Amazon
New York, NY, USA

B. Aditya Prakash
badityap@cc.gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Dmitry Efimov
defimov@amazon.com
Amazon
New York, NY, USA

Abstract

Multi-horizon time series forecasting has many practical applications such as demand forecasting. Accurate demand prediction is critical to help make buying and inventory decisions for supply chain management of e-commerce and physical retailers, and such predictions are typically required for future horizons extending tens of weeks. This is especially challenging during high-stake sales events when demand peaks are particularly difficult to predict accurately. However, these events are important not only for managing supply chain operations but also for ensuring a seamless shopping experience for customers. To address this challenge, we propose Temporal-Aligned Transformer (TAT), a multi-horizon forecaster leveraging apriori-known context variables such as holiday and promotion events information for improving predictive performance. Our model consists of an encoder and decoder, both embedded with a novel Temporal Alignment Attention (TAA), designed to learn context-dependent alignment for peak demand forecasting. We conduct extensive empirical analysis on two large-scale proprietary datasets from a large e-commerce retailer. We demonstrate that TAT brings up to 30% accuracy improvement on peak demand forecasting while maintaining competitive overall performance compared to other state-of-the-art methods.

CCS Concepts

• **Applied computing** → **Supply chain management**.

*This work was done when affiliated with Amazon, USA.

KDD '25, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/XXXXXX.XXXXXX

Keywords

Time-Series Forecasting, Demand Forecasting, Peak Prediction

ACM Reference Format:
Zhiyuan Zhao*, Sitan Yang*, Kin G. Olivares, Boris N. Oreshkin, Stan Vitebsky, Michael W. Mahoney, B. Aditya Prakash, and Dmitry Efimov. 2025. TAT: Temporal-Aligned Transformer for Multi-Horizon Peak Demand Forecasting. In *Proceedings of the 1st Workshop on "AI for Supply Chain: Today and Future" @ 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 Introduction

Time series forecasting is a fundamental problem that finds broad applications in supply chain management, finance, healthcare, and more [1, 9, 23]. In modern forecasting problems, predictions are often required over multiple time horizons, a task known as multi-horizon forecasting [4, 33]. Recent advances in deep learning time series models have achieved notable success, exemplified by recurrent neural networks and transformer-based architectures [14, 25, 28, 42, 43]. These methods have already demonstrated state-of-the-art predictive performance for producing accurate forecasts of horizons up to tens of weeks in the future.

As a practical motivation for this work, we consider industrial-scale demand prediction at a large e-commerce retailer, where forecasts are essential to inform critical supply chain decisions [2, 3]. One key challenge is to accurately predict demand patterns during peak sales events, a problem becoming increasingly important as the number and scale of such events have grown dramatically in recent years. Marked by an outstanding concentration of merchandising discounts and promotions, these peak events differ distinctively from typical factors like seasonality or product life cycles, resulting in hard-to-predict demand spikes. Both over-forecasting and under-forecasting peak demand can be problematic, leading to poor customer experience and large financial costs.

General time series methods that only focus on univariate forecasting [28] or that treat multivariate forecasting as a channel independent setup [25] are very unlikely to correctly account for

the critical impact of key exogenous variables, such as promotion and price-related information, making them ineffective for practical demand forecasting tasks. To address these limitations, the MQ-Forecaster framework [33] was introduced with designs suited for demand forecasting. It incorporates common exogenous variables such as seasonal cycles and planned peak events, but its architecture suffers from an "information bottleneck," where the encoder transmits information to the decoder only via a single hidden state. There are many subsequent works leveraging the celebrated Transformer architecture [32] that aim to address this issue. One such example is Temporal Fusion Transformer (TFT) [16], which has been shown to achieve competitive performance for a few commonly used public datasets. However, despite being general, TFT simply concatenates all features and applies self-attention. This attention mechanism can work reasonably well for small yet simple datasets. As we show later in this paper, however, it fails to perform similarly well in real demand forecasting tasks, mainly due to the failure to correctly leverage the power of key exogenous variables.

Given the limitations and challenges, in this paper, we introduce Temporal-Aligned Transformer (TAT), a novel framework that is explicitly formulated to enhance learning between the target time series and key exogenous variables, providing the context on which the model produces the forecast. Since the main application of this paper is peak demand forecasting, we consider exogenous variables related to future peak events. Our approach proves to enhance peak demand prediction accuracy, while maintaining competitive forecasting performance during non-peak periods, compared to popular forecasting methods. The main contributions are summarized as follows:

- We introduce Temporal Alignment Attention (TAA), a novel method for learning the complex dependencies between the target time series and key exogenous variables which provide the context information, and thus enable more accurate forecasts as compared to the widely adopted self-attention mechanisms in previous methods.
- We propose Temporal-Aligned Transformer (TAT), a Transformer based model that integrates TAA within an encoder-decoder architecture and demonstrates the SOTA performance in multi-horizon time series forecasting.
- We perform comprehensive evaluations of TAT against popular forecasting methods on two substantially complex industrial demand datasets from a large e-commerce retailer, we achieve up to 30% improvements over the previously reported SOTA in peak demand prediction, while still maintaining competitive results in overall performance.

We provide more detailed related works in Appendix A.

2 Problem Formulation

We adopt the same setting as in the TFT method [16], and we formulate the demand forecasting problem as an application of multi-horizon time-series forecasting, where the objective is to predict the target time series by leveraging multiple types of features. Formally, we define the dataset $\mathcal{D}$ consisting of N samples over time steps up to T. In general, each sample includes three types of features: static (time-invariant) metadata information $x^s \in \mathbb{R}^{d_s}$ (e.g., product category); historical observed time series covariates

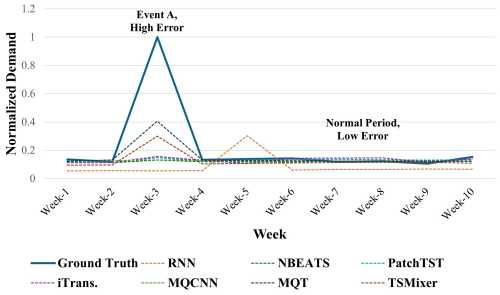

Figure 1: Week-by-week (normalized) demand for a popular book. Even with state-of-the-art forecasting models, the forecasting errors remain high during demand peaks.

$\boldsymbol{x}^b \in \mathbb{R}^{T \times d_b}$ (e.g, product detail page views); and time-varying known context about the future $\boldsymbol{x}^c \in \mathbb{R}^{(T+H) \times d_c}$ (e.g., indicators for historical and upcoming holiday dates). In line with TFT and other direct methods [4, 33], we simultaneously predict the future values of $y \in \mathbb{R}^H$ for H time steps, i.e., $h \in \{1, ..., H\}$. We incorporate static information x_s and all historical observed information $\boldsymbol{x}^b$ with a lookback window L (i.e., $\boldsymbol{x}^b_{t-L:t} = [x^b_{t-L}, \ldots, x^b_L]$). For known context time series $\boldsymbol{x}^c$, we additionally leverage its entire range of time steps (i.e., $\boldsymbol{x}^c_{t-L:t+H} = [x^c_{t-L}, \ldots, x^c_{t+H}]$). For simplicity, we denote the time range from $t - L$ to t with a down script $[L]$, the time range from $t + 1$ to $t + H$ with $[H]$, and the whole lookback and horizon range as $[L + H]$. We parametrize the model by θ, and we learn through empirical risk minimization, obtaining a function $f_\theta : (x^s, \boldsymbol{x}^b_{[L]}, \boldsymbol{x}^c_{[L+H]}) \rightarrow y_{[H]}$.

Existing forecasting approaches [4, 16, 27, 33, 35] have achieved some success in overall forecasting accuracy. However, these approaches often struggle when predicting demand peaks, particularly during holidays and promotional events, as showcased in Figure 1. In this paper, we emphasize the importance of accurately predicting demand peaks, and we highlight the critical impact of known context features, $\boldsymbol{x}^c$, in improving model performance for capturing demand patterns during peak events. These exogenous variables are broadly available, and they have been considered in previous studies [5, 16]. They are often deterministic for both historical lookback and future horizon periods, and in the industrial demand forecasting problem, $\boldsymbol{x}^c$ typically contains additional key features related to promotion events such as price discount amount and promotion type. Our main motivation is to improve peak demand forecasting accuracy by effectively leveraging x^c.

3 Temporal-Aligned Transformer

In this section, we introduce the architecture of the proposed Temporal-Aligned Transformer (TAT) model. In particular, We first describe the Temporal Alignment Attention (TAA), which provides an explicit context-dependent learning for target demand time series based on the known context features related to demand peaks. We then introduce the architecture of TAT that effectively integrates Temporal Alignment Attention in its encoder-decoder structure.

3.1 Temporal Alignment Attention

We introduce Temporal Alignment Attention (TAA), a module designed to temporally align demand patterns with known contextual variables related to demand peaks, while also capturing their interactions within both the encoder and decoder through attention mechanisms. Specifically, we consider x^c, the known contextual information comprising various holiday-related and promotion-related features, and we partition its time steps into two components: the historical part, $x^c_{[L]} \in \mathbb{R}^{L\times d_c}$, which is set to align with observed demand patterns $x^b_{[L]}$, and the apriori-known future horizon context, $x^c_{[H]} \in \mathbb{R}^{H\times d_c}$, which aligns with the forecasting horizon. This explicit separation is rarely addressed in existing demand forecasting methods.

To achieve this temporal alignment, we leverage attention mechanisms in both the encoder and decoder with inputs from e, the latent embedding space. We denote the output of any input variable through the latent space with e, e.g., $\text{embed}(x^b_{[L]}) = e^b_{[L]}$. This alignment is achieved using the standard multi-head scaled dot-product attention [32], i.e., $\text{TAA}(Q,K,V) = \text{MultiHead}(Q,K,V)$. Nevertheless, we propose using selective features for attention inputs, rather than adopting the identical Q, K, and V setup of vanilla self-attention, as commonly used in previous forecasting models. In the encoder, we choose the embedded observed time series $e^b_{[L]}$ as Q to align with the embedded historical information of context variables $e^c_{[L]}$. For finer-grained distinctions, we include in K the embedded static features e^s as metadata to differentiate demand characteristics across products. For example, some products such as routine utilities can be less sensitive to promotional events compared to items like electronic devices, a distinction that can be explicitly made by incorporating the metadata. Lastly, we include all features in V to enable additional cross-series interactions to the attention output. Consequently, the Q, K, and V setup of TAA in the encoder is defined as follows:

$$\begin{aligned} Q &= \text{Linear}(e^b_{[L]}),\\ K &= \text{Linear}([e^c_{[L]}, \text{Broadcast}(e^s)]),\\ V &= \text{Linear}([e^b_{[L]}, e^c_{[L]}, \text{Broadcast}(e^s)]),\\ &\text{Score}(Q,K) \in \mathbb{R}^{L\times L}. \end{aligned} \tag{1}$$

Here, $[\cdot]$ denotes concatenation along the hidden dimension, resulting in different dimensionalities for Q, K, and V. We then project all tensors onto a unified hidden dimension d_{hidden} using linear layers for attention computation. Additionally, the static features e^s are broadcasted to match the temporal dimension of the other input sequences.

Similarly, in the decoder, we use the initialization sequence $e^{\text{Dec}}_{[H]}$, derived from the encoder output, as Q to represent demand forecasts across all future horizons. These forecasts are then aligned with the embedded known future context variables $e^c_{[H]}$ through an additional TAA mechanism, with the Q, K, and V setup defined as:

$$\begin{aligned} Q &= \text{Linear}(e^{\text{Dec}}_{[H]}),\\ K &= \text{Linear}([e^c_{[H]}, \text{Broadcast}(e^s)]),\\ V &= \text{Linear}([e^b_{[H]}, e^c_{[H]}, \text{Broadcast}(e^s)]),\\ &\text{Score}(Q,K) \in \mathbb{R}^{H\times H}. \end{aligned} \tag{2}$$

The primary objective is to enhance context alignment within the attention mechanism by designing Q, K, and V. This enables the network to learn temporal alignment in both lookback and horizon windows by assigning greater weights to specific positions, namely those where x^b exhibits demand peaks and contextual features indicate relevant signals. In contrast, TFT [16] uses standard multi-head self-attention across all types of feature embeddings. This can be too general to capture the intricate interactions during peak events. MQTransformer [4] has a similar attention module for context features x^c. However, it compresses x^c along the temporal dimension and only applies attention between the observed target time series and the future known horizon context, causing misalignment between these features.

3.2 TAT Model Architecture

The model architecture of TAT is shown in Figure 2 and consists of the following components: the input feature embedding module; the encoder module; the encoder-decoder translation module; the decoder module; and the posterior calibration module. Each module performs distinct functions sequentially. The feature embedding module (Section 3.2.1) applies different yet effective embeddings for various feature types. This is followed by the encoder (Section 3.2.2), which incorporates a TAA together with self-attention to effectively align and enhance the learning between the target time series and promotion-related features. Next, an encoder-decoder translation (Section 3.2.3) with self-attention generates the decoder sequence based on historical patterns. Then, the decoder applies TAA and self-attention similar to those in the encoder. Finally, posterior calibration (Section 3.2.4) further refines the predictions during demand peaks. The overall model architecture is illustrated in Figure 2, and the details of each component are provided in the following subsections.

3.2.1 ***Input Feature Embedding.*** We apply different strategies to transform each type of input feature into embedding vectors, as shown in Figure 3. For static features $x^s \in \mathbb{R}^s$, we use categorical embedding to map them into a high-dimensional latent space, followed by dropout and a linear projection to a fixed hidden size. Subsequently, we broadcast them along the temporal dimension to be compatible with temporal features. For the observed historical time series $x^b_{[L]} \in \mathbb{R}^{L\times d_b}$, we apply token embedding via standard 1D dilated convolution layers [31] across time steps, followed by embedding strategies for time series, as described in previous work [43]. For known context features $x^c_{[L+H]} \in \mathbb{R}^{(L+H)\times d_c}$, we split these time series into historical ($x^c_{[L]} \in \mathbb{R}^{L\times d_c}$) and future ($x^c_{[H]} \in \mathbb{R}^{H\times d_c}$) components. We then perform separate token embedding as on each part, similar to the approach used for x^b.

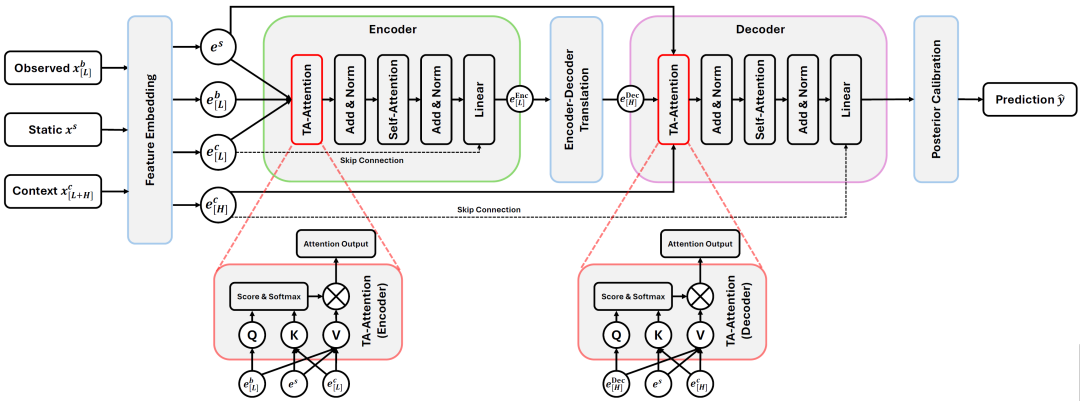

Figure 2: TAT model architecture, with the proposed TAA highlighted in red.

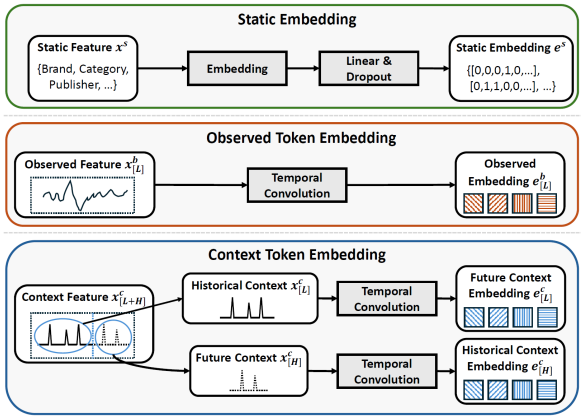

Figure 3: Diagram of the input feature embedding process for static, observed, and known context features.

3.2.2 ***Encoder-Decoder Structure.*** TAT follows an encoder-decoder architecture, as does a vanilla Transformer [32], but it employs a customized attention module in both the encoder and decoder. This module consists of two attention layers: TAA; followed by a regular self-attention. TAA first aligns the exogenous context promotion-related features with the target time series (as described in Section 3.1); and the self-attention uses the identical query, key, and value derived from the TAA outputs, to refine the aligned representations, thereby improving the overall representation quality. To ensure stable training for the attention mechanisms, we apply layer normalization and residual connections after the attention outputs, same as the vanilla Transformer [32] and Informer [43]. The entire processing module described above forms the core structure of both the encoder and decoder of TAT.

3.2.3 ***Encoder-Decoder Translation.*** To generate the decoder initialization sequence (i.e., the embedding of the decoder's input), $e^{\text{Dec}}_{[H]}$, from the encoder output tokens, we employ an encoder-decoder translation module to map the encoder output to the decoder input, achieved by a transposed self-attention. This self-attention uses the transposed encoder tokens as input, with the temporal dimension placed as the final dimension. The self-attention approach treats each hidden dimension as a token and predicts the future tokens through modeling channel-wise dependencies, following a methodology similar to iTransformer [21]. It then has a linear projection layer to adjust the token size from the lookback window size to the horizon window size, generating the initial latent representation for the decoder. The full operation is as follows:

$$\begin{aligned} e^{\text{Enc}}_{[L]} &= \text{Encoder}(e^b_{[L]}, e^c_{[L]}, e^s) \in \mathbb{R}^{L\times d_{hidden}} \\ e^{\text{Dec}}_{[H]} &= \text{Linear}(\text{Self-Attention}(Q,K,V))^T \in \mathbb{R}^{H\times d_{hidden}} \\ &\text{where } Q,K,V = (e^{\text{Enc}}_{[L]})^T, \end{aligned} \tag{3}$$

where $e^{\text{Enc}}_{[L]}$ denotes the encoded tokens from the encoder output.

Previous studies [21] typically treat $e^{\text{Dec}}_{[H]}$ as the final predictions, but we propose leveraging these predictions as the decoder initialization. This approach further enhances the alignment between predicted patterns and the known future context variables $x^c_{[H]}$, allowing refinement through temporal alignment in the decoder. Moreover, compared to the commonly used zero decoder initialization in time-series forecasting [43], the encoder-decoder self-attention enables more effective use of historical patterns, and also avoid the information bottleneck described in [4] due to transmitting from encoder to decoder via a single hidden state. By reshaping and projecting past information onto future horizons, this mechanism generates a more effective decoder initialization sequence.

3.2.4 ***Posterior Calibration.*** Besides the model accuracy, calibration issues often arise during demand peaks, as it is generally hard to predict the magnitude of impact on demand with a concentration of drivers. Therefore, in TAT, we design an additional step referred to as the Posterior Calibration module, as follows:

$$\begin{aligned} y^{\text{Dec}}_{[H]} &= \text{Decoder}(e^{\text{Dec}}_{[L]}, e^c_{[H]}, e^s) \\ \hat{y} &= y^{\text{Dec}}_{[H]} * (1 + \text{Calib}(x^c_{[H]})), \end{aligned} \tag{4}$$

where $y^{\text{Dec}}_{[H]}$ is the decoder's output, $\hat{y}$ is the final calibrated prediction, and $\text{Calib}(\cdot)$ is a simple multi-layer perceptron (MLP) serving

as the calibration layer. This layer takes future features x^c as input and outputs a scaling factor to adjust the decoder's predictions. The rationale behind the calibration is that predictions during demand peaks are generally more challenging compared to normal periods. Posterior calibration aims to rescale predictions at demand peaks, while preserving the accuracy of predictions for normal periods.

3.2.5 ***Loss Function.*** TAT is trained by jointly minimizing a set of quantile losses. Consistent with previous studies, we evaluate the 50^{th} and 90^{th} quantile losses (unweighted P_{50} and P_{90}), commonly used in forecasting to provide predictions with associated uncertainty and is employed in popular literature [4, 33, 39]. The overall loss function is formulated as follows:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{P50}(y, \hat{y}^1) + \mathcal{L}_{P90}(y, \hat{y}^2) \\ &= \sum \left[0.5 \max(y - \hat{y}^1, 0) + 0.5 \max(\hat{y}^1 - y, 0)\right] \\ &+ \sum \left[0.9 \max(y - \hat{y}^2, 0) + 0.1 \max(\hat{y}^2 - y, 0)\right]\end{aligned} \tag{5}$$

where y represents the ground truth values, $\hat{y}^1$ denotes the 50^{th} quantile predictions, and $\hat{y}^2$ corresponds to the 90^{th} quantile predictions. TAT produces two-dimensional outputs from its final linear layer: the first dimension provides the 50^{th} quantile predictions ($\hat{y}^1$); and the second dimension provides the 90^{th} quantile predictions ($\hat{y}^2$). The training objective jointly optimizes the loss by summing both quantile-specific losses.

4 Empirical results

4.1 Setup

4.1.1 Dataset. We evaluate TAT by comparing its performance against popular time-series forecasting baselines on two large-scale retail demand datasets from a large e-commerce retailer, namely **E-Commerce Retail Datasets**. We obtain a large-scale proprietary demand data consisting of worldwide products sold on a large e-commerce retailer with demand time series for hundreds of thousands of products spanning more than 10 years. These time series are in weekly grain and compounded with various exogenous features. With the data, we further select two smaller datasets, namely **E-Commerce Retail-Region 1** and **Region 2**, with products from two main geographic locations, where demand patterns can be quite different even for the same product. We use the 4-year period data for training and the following last year for testing. More detailed dataset statistics are in Table 3, Appendix B.

4.1.2 Baselines. For general time-series forecasting baselines, we include conventional RNN [6] and N-BEATS [26, 28], and most recent SOTA forecasting models PatchTST [25], iTransformer [21] (iTrans.), and TSMixer [5]. (Univariate, multivariate, and univariate with exogenous features models, respectively.) For industrial demand forecasting baselines, we include MQCNN [33], TFT [16], and MQTransformer (MQT) [4]. Other well-known baselines, such as ARIMA [10], DeepAR [29], and ConvTrans [15], are omitted since their performances are less competitive than the selected baselines shown by previous studies [4, 16].

4.1.3 Evaluation Metrics. We evaluate the performance of the baselines and our proposed TAT using two metrics as described below. The first metric is the standard *overall accuracy* (Table 2), where we use the weighted 50^{th} and 90^{th} percentile quantile loss (P_{50} and P_{90}) [4], defined as follows:

$$P_\alpha = 2\frac{\sum_{y_{i,t}\in\tilde{\Omega}} \sum_{h=1}^{H} \mathcal{L}_\alpha(y_{i,t}, \hat{y}(\alpha, h))}{\sum_{y_{i,t}\in\tilde{\Omega}} \sum_{h=1}^{H} |y_{i,t}|}, \quad \alpha = 50, 90. \tag{6}$$

where Ω is the domain of training data containing all N samples and $\mathcal{L}_\alpha$ is the quantile loss function described in Equation (5).

In addition to overall accuracy, We also consider *target date accuracy* (Table 1) as an important metric as it measures the model performance directly towards the peak events of interest. In this case, P_{50} and P_{90} are defined as follows:

$$P_\alpha = 2\frac{\sum_{y_{i,t}\in\tilde{\Omega}} \sum_{h=1}^{H} \mathcal{L}_\alpha(y_{i,t}, \hat{y}(\alpha, h)) \cdot \mathbb{1}_{\{t+h=\text{E}\}}}{\sum_{y_{i,t}\in\tilde{\Omega}} \sum_{h=1}^{H} |y_i|}, \quad \alpha = 50, 90. \tag{7}$$

Here, E represents a peak event, and the metric is specifically measured against forecasts generated h weeks apart from the target date, i.e., the event date. In this paper, we consider two major yearly promotional events, denoted as Event A and Event B. We omit the normalized mean absolute error (NMAE), as it is similar to the P_{50} quantile loss. Other evaluation metrics, such as normalized mean square error (NMSE), are omitted, as they receive less interest and are not employed when evaluating both previous E-Commerce Retail studies [4].

4.1.4 Reproducibility. All models are trained using eight NVIDIA Tesla V100 32GB GPUs. Detailed information on implementation, hyperparameter tuning, and training efficiency is provided in Appendix B. All evaluations are averaged over three independent runs or random seeds, and the standard deviation is reported in Appendix C for methods with close performance. For privacy considerations, results on the E-Commerce Retail Datasets have been normalized relative to RNN results (e.g., RNN results are set to 1.0, and other results are scaled accordingly).

4.2 Demand Forecasting Results

Here, we present the evaluations of forecasting results for TAT and baseline models on E-Commerce Retail Datasets. Since the primary objective is to evaluate peak forecasting performance, we pick two major sales events shared by two datasets, and we compare all models' performance in terms of target date accuracy (equation (7)) by setting the target date as an event date. In addition, we also compare overall accuracy across all models.

4.2.1 Peak forecast accuracy. We evaluate the forecasting performance over promotional events. We fix the forecast generation date and evaluate model performance in terms of the target date accuracy, and we focus on two events, Event A and B, occurring at the 3^{rd} and 23^{rd} horizons, respectively. This evaluation follows a standard time-series forecasting test paradigm [16]. The evaluation results are summarized in Table 1. The results show clear advantages of TAT in improving peak demand forecasting accuracy, particularly for shorter horizons. TAT achieves up to 10% lower P_{50} error and 20% lower P_{90} error for Event A and 5% lower P_{50} and P_{90} errors on Event B. Furthermore, the performance improvements in E-Commerce Retail-Region 1 are more significant than in Region 2. This can be attributed to the fact that E-Commerce Retail-Region 1

Table 1: Evaluations with fixed forecast generation date for two main peak events on two proprietary demand datasets. The best results are bold, and the second-best results are underlined. All values indicate relative performances versus the RNN result.

E-Commerce Retail-Region 1										
Event	Horizon	Metric	RNN	N-BEATS	PatchTST	iTrans.	MQCNN	TFT	MQT	TSMixer	TAT
Event A	3	P50	1.000	0.685	0.667	0.674	0.700	0.623	0.602	0.632	**0.551**
		P90	1.000	0.551	0.505	0.520	0.651	0.454	0.380	0.445	**0.309**
Event B	23	P50	1.000	0.768	0.746	0.835	0.742	0.759	0.740	0.736	**0.717**
		P90	1.000	0.582	0.570	0.645	0.552	0.650	0.556	0.541	**0.529**
E-Commerce Retail-Region 2										
Event	Horizon	Metric	RNN	N-BEATS	PatchTST	iTrans.	MQCNN	TFT	MQT	TSMixer	TAT
Event A	3	P50	1.000	1.100	0.892	0.923	0.918	0.865	0.835	0.935	**0.830**
		P90	1.000	0.766	0.635	0.690	0.770	0.585	0.554	0.665	**0.503**
Event B	23	P50	1.000	0.951	0.542	0.544	0.534	0.884	0.528	0.800	**0.513**
		P90	1.000	0.894	0.761	0.766	0.641	0.912	**0.639**	0.776	0.659

Table 2: Overall accuracy metrics aggregated across all horizons on two proprietary demand datasets. The best results are bold, and the second-best results are underlined. All values are relative performances to the RNN result.

E-Commerce Retail-Region 1									
Metric	RNN	N-BEATS	PatchTST	iTrans.	MQCNN	TFT	MQT	TSMixer	TAT
P50	1.000	0.669	0.705	0.643	0.624	0.653	**0.611**	0.653	0.617
P90	1.000	0.520	0.542	0.520	0.479	0.538	0.452	0.479	**0.443**
E-Commerce Retail-Region 2									
Metric	RNN	N-BEATS	PatchTST	iTrans.	MQCNN	TFT	MQT	TSMixer	TAT
P50	1.000	0.999	0.645	0.659	0.630	0.896	0.624	0.880	**0.617**
P90	1.000	0.869	0.745	0.766	0.675	0.840	**0.669**	0.769	0.681

exhibits sharper demand peaks during these events, where TAT is expected to be more effective.

4.2.2 Overall forecast accuracy. In addition to the performance gains on forecasting demand peaks, we also evaluate the performance of TAT to show competitive performance compared to state-of-the-art baselines on overall forecasting accuracy across all horizons. We follow standard time-series forecasting setups [16, 43], where predictions are made at a fixed date, same as Table 1, and extend up to 26 weeks horizons. The results that are aggregated across all forecasting horizons are summarized in Table 2. The proposed TAT demonstrates competitive performance compared to state-of-the-art demand forecasting models, achieving leading results on P_{50} errors and comparable performance on P_{90} errors for both the E-Commerce Retail-Region 1 and Region 2 datasets.

4.3 Ablation Study

We conduct an ablation study to evaluate the contribution of each module in TAT. Specifically, we compare the performance of TAT with three ablated variants: TAT\TAA, which excludes the Temporal Alignment Attention in both encoder and decoder; TAT\SA, which excludes the self-attention in both encoder and decoder; and TAT\PC, which excludes the posterior calibration. The evaluations are performed on the E-Commerce Retail-Region 1 dataset, analyzing both the overall performance and the performance during Event A. The results of the ablation study are presented in Figure 4.

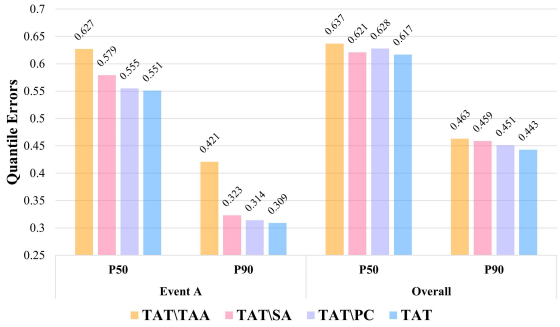

Figure 4: Ablation study on TAT and its three variants. The results demonstrate the effectiveness of the proposed Temporal Alignment Attention, which achieves incremental improvements of 12% in P50 and 26% in P90 during Event A. Furthermore, removing any module from TAT results in a performance decline, underscoring the importance of other modules to the effectiveness of TAT.

The results of the ablation study demonstrate the effectiveness of each module in TAT, as evidenced by the performance degradation observed in all ablated variants, where one module is removed. Notably, excluding the Temporal Alignment Attention module (TAT\TAA) results in the most significant performance drop for Event A demand peak predictions, with approximately 15% higher P_{50} error and 40% higher P_{90} error, in spite of the overall

performance remains relatively close to that of TAT. This finding highlights the critical role of the proposed Temporal Alignment Attention module in aligning and modeling the correlations between future features and demand time series, thereby enhancing the accuracy of peak demand predictions.

5 Conclusion

In this work, we have proposed a novel transformer-based framework, TAT, aimed at improving demand forecasting accuracy during demand peaks by effectively leveraging known context features. TAT uses a novel attention mechanism, Temporal Alignment Attention, which aligns and models the correlations between future features and demand time series during peak periods, resulting in more accurate predictions. In addition, the newly proposed posterior calibration in TAT further enhances performance by directly calibrating decoder outputs and known context features. Empirical evaluations demonstrate the effectiveness of TAT in improving forecasting accuracy during demand peaks, while maintaining competitive performance compared to state-of-the-art models in terms of overall forecasting accuracy.

6 Acknowledgment

This work was done while the first author was a full-time and part-time intern at Amazon. The project was primarily supported by Amazon. We also acknowledge partial support from the NSF (Expeditions CCF-1918770, CAREER IIS-2028586, Medium IIS-1955883, Medium IIS-2106961, Medium IIS-2403240, PIPP CCF-2200269), CDC MInD program, Meta faculty gift, and funds/computing resources from Georgia Tech and GTRI.

References

[1] M Zied Babai, John E Boylan, and Bahman Rostami-Tabar. 2022. Demand forecasting in supply chains: a review of aggregation and hierarchical approaches. *International Journal of Production Research* 60, 1 (2022), 324–348.

[2] Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. 2017. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1694–1705.

[3] Frank Chen, Zvi Drezner, Jennifer K Ryan, and David Simchi-Levi. 2000. Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management science* 46, 3 (2000), 436–443.

[4] Kevin C. Chen, Lee Dicker, Carson Eisenach, and Dhruv Madeka. 2022. MQ-Transformer: Multi-horizon forecasts with context dependent attention and optimal bregman volatility. In *KDD 2022*, Vol. Workshop on Mining and Learning from Time Series – Deep Forecasting: Models, Interpretability, and Applications. Association for Computing Machinery, New York, NY, United States., - pages. https://www.amazon.science/publications/mqtransformer-multi-horizon-forecasts-with-context-dependent-attention-and-optimal-bregman-volatility

[5] Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. 2023. TSMixer: An All-MLP Architecture for Time Series Forecast-ing. *Transactions on Machine Learning Research* - (2023), –. https://openreview.net/forum?id=wbpxTuXgm0

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *28th International Conference on Neural Information Processing Systems (NIPS 2014)*. Curran Associates Inc., Montreal, Canada.

[7] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688* (2023).

[8] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 7522–7529.

[9] Clive William John Granger and Paul Newbold. 2014. *Forecasting economic time series*. Academic Press, United States.

[10] Rob J Hyndman, George Athanasopoulos, Azul Garza, Cristian Challu, Max Mergenthaler, and Kin G. Olivares. 2024. *Forecasting: Principles and Practice, the Pythonic Way*. OTexts, Melbourne, Australia. available at https://otexts.com/fpppy/.

[11] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *12th International Conference on Learning Representations*. ICLR 2024, Vienna, Austria, –. https://openreview.net/forum?id=Unb5CVPtae

[12] Harshavardhan Kamarthi and B Aditya Prakash. 2023. Large Pre-trained time series models for cross-domain Time series analysis tasks. *arXiv preprint arXiv:2311.11413* (2023).

[13] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*.

[14] 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*. Association for Computing Machinery, Ann Arbor, Michigan, USA, 95–104.

[15] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems* 32 (2019).

[16] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* 37, 4 (2021), 1748–1764.

[17] Haoxin Liu, Harshavardhan Kamarthi, Lingkai Kong, Zhiyuan Zhao, Chao Zhang, and B Aditya Prakash. 2024. Time-series forecasting for out-of-distribution generalization using invariant learning. *arXiv preprint arXiv:2406.09130* (2024).

[18] Haoxin Liu, Chenghao Liu, and B Aditya Prakash. 2024. A Picture is Worth A Thousand Numbers: Enabling LLMs Reason about Time Series via Visualization. *arXiv preprint arXiv:2411.06018* (2024).

[19] Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, et al. 2024. Time-MMD: Multi-Domain Multimodal Dataset for Time Series Analysis. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems NeurIPS 2024*, Vol. Datasets and Benchmarks Track. NeurIPS 2024, Vancouver, Canada, –.

[20] Haoxin Liu, Zhiyuan Zhao, Jindong Wang, Harshavardhan Kamarthi, and B. Aditya Prakash. 2024. LSTPrompt: Large Language Models as Zero-Shot Time Series Forecasters by Long-Short-Term Prompting. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 7832–7840. https://doi.org/10.18653/v1/2024.findings-acl.466

[21] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted transformers are effective for time series forecasting. In *12th International Conference on Learning Representations*. ICLR 2024, Vienna, Austria, –. https://openreview.net/forum?id=JePfAI8fah

[22] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. 2023. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems* 36 (2023), 14273–14292.

[23] Sarabeth M Mathis, Alexander E Webber, Tomás M León, Erin L Murray, Monica Sun, Lauren A White, Logan C Brooks, Alden Green, Addison J Hu, Roni Rosenfeld, et al. 2024. Title evaluation of FluSight influenza forecasting in the 2021–22 and 2022–23 seasons with a new target laboratory-confirmed influenza hospitalizations. *Nature Communications* 15, 1 (2024), 6289.

[24] Elizbar A Nadaraya. 1964. On estimating regression. *Theory of Probability & Its Applications* 9, 1 (1964), 141–142.

[25] 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 (ICLR 2023)*. ICLR, Kigali Rwanda.

[26] Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting* 39, 2 (2023), 884–900. https://doi.org/10.1016/j.ijforecast.2022.03.001

[27] Kin G. Olivares, O. Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. 2024. Probabilistic hierarchical forecasting with deep Poisson mixtures. *International Journal of Forecasting* 40, 2 (2024), 470–489. https://doi.org/10.1016/j.ijforecast.2023.04.007

[28] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations*. ICLR 2020, Addis Ababa, Ethiopia, –. https://openreview.net/forum?id=r1ecqn4YwB

[29] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting* 36, 3 (2020), 1181–1191.
[30] Alex J Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and computing* 14, 3 (2004), 199–222.
[31] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR* abs/1609.03499 (2016). arXiv:1609.03499 http://arxiv.org/abs/1609.03499
[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.
[33] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2017. A Multi-Horizon Quantile Recurrent Forecaster. In *31st Conference on Neural Information Processing Systems NIPS 2017*, Vol. Time Series Workshop. Curran Associates Inc., Long Beach California USA, –. arXiv:1711.11053 [stat.ML] https://arxiv.org/abs/1711.11053
[34] Christopher Williams and Carl Rasmussen. 1995. Gaussian processes for regression. *Neural information processing systems (NIPS 1995)* 8 (1995), - pages.
[35] Malcolm Wolff, Kin G. Olivares, Boris Oreshkin, Sunny Ruan, Sitan Yang, Abhinav Katoch, Shankar Ramasubramanian, Youxin Zhang, Michael W. Mahoney, Dmitry Efimov, and Vincent Quenneville-Bélair. 2024. ♠ SPADE ♠ Split Peak Attention DEcomposition. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems NeurIPS 2024*, Vol. Time Series in the Age of Large Models Workshop. NeurIPS 2024, Vancouver, Canada, –. arXiv:2411.05852 [cs.LG] https://arxiv.org/abs/2411.05852
[36] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.
[37] Sitan Yang, Carson Eisenach, and Dhruv Madeka. 2022. MQRetNN: Multi-Horizon Time Series Forecasting with Retrieval Augmentation. arXiv:2207.10517 [cs.LG] https://arxiv.org/abs/2207.10517
[38] Sitan Yang, Malcolm Wolff, Shankar Ramasubramanian, Vincent Quenneville-Belair, Ronak Metha, and Michael W. Mahoney. 2023. GEANN: Scalable Graph Augmentations for Multi-Horizon Time Series Forecasting. arXiv:2307.03595 [cs.LG] https://arxiv.org/abs/2307.03595
[39] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems* 29 (2016).
[40] Hanyu Zhang, Chuck Arvin, Dmitry Efimov, Michael W Mahoney, Dominique Perrault-Joncas, Shankar Ramasubramanian, Andrew Gordon Wilson, and Malcolm Wolff. 2024. LLMForecaster: Improving Seasonal Event Forecasts with Unstructured Textual Data. In *Thirty-Eighth Annual Conference on Neural Information Processing Systems NeurIPS 2024*, Vol. Time Series in the Age of Large Models Workshop. NeurIPS 2024, Vancouver, Canada, –. https://arxiv.org/abs/2412.02525
[41] Zhiyuan Zhao, Haoxin Liu, Alexander Rodriguez, and B. Aditya Prakash. 2025. Performative Time-Series Forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
[42] Zhiyuan Zhao, Juntong Ni, Shangqing Xu, Haoxin Liu, Wei Jin, and B Aditya Prakash. 2025. TimeRecipe: A Time-Series Forecasting Recipe via Benchmarking Module Level Effectiveness. *arXiv preprint arXiv:2506.06482* (2025).
[43] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35-12. Association for the Advancement of Artificial Intelligence, Vancouver, Canada, 11106–11115.
[44] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*. PMLR, Baltimore, Maryland USA, 27268–27286.

A Related Work

Time-Series Forecasting. Classical statistical time-series forecasting models, such as ARIMA [10], often face limitations in capturing complicated patterns and dependencies due to inherent model constraints [24, 30, 34]. Recent work that applied machine learning to time-series forecasting using RNNs, LSTNet, N-BEATS has achieved notable improvements in accuracy [6, 14, 26, 28]. State-of-the-art models build upon the successes of transformer-based architectures [32], such as Informer, Autoformer, Fedformer, PatchTST, iTransformer [21, 25, 36, 43, 44], have shown significantly improved forecasting accuracy. In parallel, a growing body of work explores model-agnostic strategies to enhance out-of-distribution generalization in time-series forecasting, further boosting performance independently of the underlying forecasting model [8, 13, 17, 22, 41].

Demand Forecasting. General time-series forecasting has achieved significant success, but demand forecasting is a variant of time-series forecasting that requires a setup that differs from standard approaches. Specifically, it involves predicting targets by leveraging exogenous features, some (but not all) of which are known in advance. Early efforts focused on conventional recurrent models [27, 33], while recent advancements explored transformer-based models and foundation models, achieving notable improvements in forecasting accuracy [4, 7, 12, 16, 37, 38]. Additionally, the rise of multimodal approaches in time-series forecasting [11, 18–20] has inspired methods that integrate unstructured text data for demand forecasting [40]. However, these approaches [16, 40] often face challenges in data and computational scalability, making it difficult to apply them effectively to real-world, large-scale demand forecasting datasets.

B Empirical Evaluation Detail

In this section, we provide additional details on our empirical evaluation.

B.1 Dataset Statistics

We here present the dataset statistics for the dataset used in this work, including two industrial datasets from the e-commerce retailer. The statistics are shown in Table 3.

Table 3: Statistics of the datasets used in the evaluations. "Unique" indicates that each sample is from distinct entities with no overlap.

Dataset	Region 1	Region 2
Train Samples	270k	280k
Test Samples	280k	340k
Entities	Unique	Unique
Lookback Window	208	208
Forecast Horizon	26	26

B.2 Model Implementation

For the industrial e-commerce retail datasets, we set the forecasting horizon window size to $H = 26$ (half year), aligning with industrial requirements and interests [4, 33]. The lookback window size L varies depending on the model. For MQCNN and MQT, we use the full historical data ($L = 208$) as the lookback window. For other sequential models, we use a shorter lookback window of $L = 104$. This difference is based on the characteristics of the dataset and models. Specifically, certain demand time series exhibit low values at earlier stages that rise sharply later. Using an excessively long lookback window for sequential models may cause instability in fitting these cases. Consequently, a shorter lookback window is preferred for these models. In contrast, MQCNN and MQT mitigate

Table 4: TAT's hyperparameters for E-Comm. datasets.

Hyperparameter	Region 1	Region 2
Hidden	60	60
Dropout (x_s)	0.5	0.5
Dropout (Other)	0.1	0.1
Heads	1	1
Optimizer	AdamW	AdamW
Batchsize	512	512
Learning Rate	1e-3	1e-3
Epochs	100	50
Earlystop	✗	✗

this limitation by leveraging importance sampling, which allows them to effectively use full historical data ($L = 208$).

For baseline implementations, we follow the publicly available implementations provided by established libraries. Specifically, we use the N-BEATS implementation from the repository available at https://github.com/philipperemy/n-beats, the TFT implementation fromhttps://github.com/google-research/google-research/tree/master/tft, and the transformer-based model implementations from Time-Series Library at https://github.com/thuml/Time-Series-Library. These repositories offer reliable implementations, ensuring consistency and reproducibility in baseline comparisons.

B.3 Hyperparameter Tuning

We tune the model hyperparameters following different strategies on each dataset, ensuring alignment with the specific characteristics of the data. For the E-Commerce Retail datasets, we follow the common practices established in prior studies [4]. Specifically, we employ batch sizes of 64 and epochs of 10 on MQCNN and MQT with importance sampling, and batch size 512 and epochs of 50 or 100 epochs for other sequential models without importance sampling. Additional hyperparameters such as hidden size and number of layers are fine-tuned using a small subset of the training data.

The hyperparameters used in TAT are detailed in Table 4. All experiments are conducted using three random seeds or three independent runs, and the reported results are averaged across these runs.

B.4 Training Efficiency

The training efficiency on the E-Commerce Retail Datasets of selected models used in our experiments is reported in Table 5. Models with undesirable performance, such as RNN, are excluded from this analysis due to their limited relevance and lower practical interest. As shown in the table, TAT demonstrates reasonable training efficiency compared to other baseline models. While it trains slightly slower than general time-series forecasting models like TSMixer, it significantly outperforms practical demand forecasting models such as MQT (which also gives more accurate predictions in most cases) in terms of training speed.

Table 5: Training efficiency between TAT and baselines on E-Commerce Retail Datasets.

Method	Batchsize	Epoch	Efficiency
PatchTST	512	100	0.5x
TSMixer	512	100	0.4x
TFT	64	50	11.4x
MQCNN	64	100	2.3x
MQT	64	100	9.0x
TAT	512	50 (E) 100 (A)	1.0x (~8 sec/epoch)

Table 6: Standard deviations for TAT and baseline models with close performance on E-Commerce Retail Datasets (Performance on demand peaks with fixed prediction date, extended Table 1).

E-Commerce Retail-Region 1					
Event	Metric	MQCNN	MQT	TSMixer	TAT
A	P50	0.700 (±0.0066)	0.602 (±0.0130)	0.632 (±0.0121)	**0.551** (±0.0132)
	P90	0.651 (±0.0068)	0.380 (±0.0184)	0.445 (±0.0129)	**0.309** (±0.0198)
B	P50	0.742 (±0.0053)	0.740 (±0.0128)	0.736 (±0.0072)	**0.717** (±0.0115)
	P90	0.552 (±0.0053)	0.556 (±0.0180)	0.541 (±0.0130)	**0.529** (±0.0098)
E-Commerce Retail-Region 2					
Event	Metric	MQCNN	MQT	TSMixer	TAT
A	P50	0.918 (±0.0008)	0.835 (±0.0147)	0.935 (±0.0165)	**0.830** (±0.0020)
	P90	0.770 (±0.0049)	0.554 (±0.0180)	0.665 (±0.0110)	**0.503** (±0.0059)
B	P50	0.534 (±0.0100)	0.528 (±0.0086)	0.800 (±0.0220)	**0.513** (±0.0068)
	P90	0.641 (±0.0155)	**0.639** (±0.0166)	0.776 (±0.0197)	0.659 (±0.0111)

Table 7: Standard deviations for TAT and baseline models with close performance on E-Commerce Retail Datasets (Averaged performance on all horizons, extended Table 2).

E-Commerce Retail-Region 1				
Metric	MQCNN	MQT	TSMixer	TAT
P50	0.624 (±0.0014)	**0.611** (±0.0036)	0.653 (±0.0140)	0.617 (±0.0040)
P90	0.479 (±0.0021)	0.452 (±0.0062)	0.479 (±0.0066)	**0.443** (±0.0053)
E-Commerce Retail-Region 2				
Metric	MQCNN	MQT	TSMixer	TAT
P50	0.630 (±0.0042)	0.624 (±0.0056)	0.880 (±0.0190)	**0.617** (±0.0064)
P90	0.675 (±0.0041)	**0.669** (±0.0139)	0.769 (±0.0048)	0.681 (±0.0100)

C Additional Results

To mitigate potential bias and noise in the results, we report the average performance of TAT and baseline models across three independent runs with different random seeds in the main results. We further report the standard deviations of MQCNN, MQT, TSMixer, and TAT of evaluations on the E-Commerce Retail Datasets (Table 1 and Table 2), due to their close performance. (See the tables below.) We choose these baselines as they have achieved the best or second-best results at least one time in the corresponding evaluations. Other baselines are excluded from these tables as their performances are uniformly lagging behind.