# RSight: A deep neural network for product demand forecasting over geographic regions

Hanjing Zhu *
hzhuad@amazon.com

Sitan Yang*
sitanyan@amazon.com

Tanmay Gupta*
tanmaygp@amazon.com

Harrison Waldon*
hwaldon@amazon.com

Kin G. Olivares*
kigutie@amazon.com

Rahul Gopalsamy*
rahulgo@amazon.com

## Abstract

We propose RSight, a new deep neural network model for product demand forecasting across multiple geographic regions. Our model employs a novel region-enhanced encoder to learn cross-regional information. Using a dataset consisting of weekly sales for 15 million products from a large e-commerce company at the US Zip2 level, our method achieves substantial accuracy improvement over existing state-of-the-art forecasting models. Additionally, we demonstrate that RSight exhibits scaling effects with data sizes as we increase the number of series in our training population, we observe substantive performance improvements.

## Keywords

Time-Series Forecasting, Demand Forecasting, Peak Prediction

**ACM Reference Format:**

Hanjing Zhu, Sitan Yang, Tanmay Gupta, Harrison Waldon, Kin G. Olivares, and Rahul Gopalsamy. 2025. RSight: A deep neural network for product demand forecasting over geographic regions. 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, 8 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

## 1 Introduction

Inventory management at scale is a critical challenge for modern retailers who handle millions of daily transactions of a catalogue consisting of billions of products. With different delivery-time promises, retailers must precisely position inventory near customer demand. While national-level demand forecasts help with aggregate buying decisions, they are insufficient for detailed retail operations such as delivery; knowing that 10,000 units of a product will be sold nationwide offers little guidance on whether to stock warehouses in Seattle or Miami to enable fast deliveries. Furthermore, point estimates of demand are insufficient for risk-aware planning, which means that retailers require forecasts of the entire distribution of regional demand in order to make optimal decisions.

The problem of probabilistic regional demand forecasting is particularly complex due to the nonlinear, spatio-temporal nature of consumer behavior. Regional demand patterns are influenced by numerous local factors including events, weather conditions, and demographic shifts. Furthermore, regions can exhibit complex interactions, with demand surges in one area impacting neighboring areas, or seasonal patterns varying across geographical regions.

Traditional approaches for demand forecasting have relied on classical time-series methods, such as ARIMA or exponential smoothing. These methods are often highly scalable, but generally fail to capture complex non-linear, spatio-temporal relationships present in regional demand data. Recent advances in deep learning, however, have shown promising results in general time-series forecasting, with deep learning models achieving state-of-the-art performance on benchmark tasks such as forecasting weather patterns, traffic, and stock prices. While there exists work using neural networks to forecast demand at a global or product level, the specific challenges of probabilistic regional demand forecasting remain largely unaddressed. Hence, there is a clear need for scalable, probabilistic forecasting models which can capture the complex spatio-temporal dynamics of regional demand. To address this gap, we introduce RSight, a deep neural network architecture specifically designed for probabilistic regional demand forecasting.

In summary, our key contributions are as follows:

(1) We present RSight, a novel deep learning architecture for probabilistic regional demand forecasting. The model introduces both 1D and 2D convolutions to learn region-specific and region-agnostic information and demonstrates state-of-the-art predictive performance.
(2) We demonstrate that RSight outperforms various state-of-the-art time-series forecasting models, achieving ~9% improvement over PatchTST [17] and NBEATS [20] in weighted quantile loss on real-world datasets
(3) We conduct comprehensive ablation studies to validate the efficacy of each component in RSight
(4) Motivated by [29], we show the data scaling effect of RSight, demonstrating our model's suitability for large-scale demand forecasting tasks.

The rest of paper is organized as follows. Section 3 formalizes the problem of regional demand forecasting. Afterwards, in Section 4.1, we present a review of MQTransformer [6], upon which our model is built. We then introduce our model, RSight, in Section 4.2. Then, experiment results and a detailed analysis are presented in Section 5, demonstrating the superior performance of RSight over state-of-the-art benchmark models. An ablation study is also presented in the same section. A concluding remark is given at the end in Section 6.

* Amazon SCOT Forecasting. New York, New York. USA.


*KDD '25, Toronto, ON, Canada.*

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

## 2 Related Work

*Deep Learning for Time-Series Forecasting.* Deep learning has become a dominant paradigm in time-series forecasting, driven in large part by the success of the Transformer architecture [22]. Originally introduced for natural language processing, Transformers have been adapted to time-series data through numerous variants which include time-series based inductive biases as well as augmentations which address bottlenecks of vanilla Transformers. Informer [32], Autoformer [26], and Fedformer [33] incorporate frequency-based decomposition to improve long-horizon forecasting. Other adaptations such as LogTrans [9] and Pyraformer [10] focus on improving locality and scalability via customized attention mechanisms. Recent models like PatchTST [17], iTransformer [11], ContiFormer [4], and the Rough Transformer [14] introduce architectural innovations including patch-based inputs, inverted attention, and continuous-time processing to further improve forecasting performance. Spectral attention mechanisms such as those in [15, 16] have been proposed as an alternative to standard dot-product attention, offering greater inductive bias for highly seasonal signals.

Alongside these Transformer-based approaches, a range of alternative deep learning architectures have shown competitive or even superior performance. Models such as N-BEATS [20], NHITS [2], and N-BEATSx [18] use feedforward networks with basis expansion to learn interpretable representations of time-series dynamics. TSMixer [3] demonstrates that all-MLP architectures can rival Transformers when structured to exploit temporal dependencies. DLinear [31] further challenges the necessity of complex models, showing that linear layers alone can serve as effective and efficient forecasters.

The aforementioned architectures are proposed as general time-series models, while other attention-based models such as MQCNN [23], MQTransformer [6] and SPADE [24] specialize in demand quantile forecasting and have demonstrated promising results. DeepAR [21] remains a foundational probabilistic model for multivariate time-series forecasting and is widely used in demand forecasting settings.

*Spatio-Temporal Forecasting.* Forecasting regional-level demand requires models that can capture spatio-temporal dynamics. A popular approach for forecasting such systems is Graph Neural Networks (GNNs). GraphCast [8] demonstrates the power of GNN-based architectures for high-resolution weather forecasting. Hierarchical and graph-based approaches have also gained traction: HiSTGNN [12] and hierarchical ST-GNNs [13] incorporate spatial hierarchies explicitly.

Hierarchical structure in regional or categorical data introduces additional modeling challenges. Probabilistic hierarchical forecasting techniques such as deep Poisson mixtures [19] aim to capture aggregate-level coherence. Other works [12, 13, 28] use graph-based encodings to model inter-regional dependencies and hierarchies across spatial and temporal scales.

*Foundation Models for Time Series.* Recently, foundation models pretrained on broad time-series data have emerged. Chronos [1], Time-LLM [7], Moirai [25], and TimesFM [5] explore the scaling laws and transfer capabilities of large models in the time-series domain. GPT4TS [34] applies pretrained language models directly to time-series tasks, highlighting the potential for unification of sequence modeling across modalities.

## 3 Problem Formulation

We consider the problem of forecasting weekly demand for a population of $N$ products. Let $Y \in \mathbb{R}^{N\times T}$ denote the time series of weekly demand for the past $T$ weeks. To forecast weekly demand, we consider four types of inputs:

(1) $\mathbf{X}^{(s)} \in \mathbb{R}^{N\times m}$: a set of $m$ static covariates,
(2) $\mathbf{X}^{(g)} \in \mathbb{R}^{N\times T\times d_g}$: product-level past historical inputs ,
(3) $\mathbf{X}^{(f)} \in \mathbb{R}^{N\times T\times d_f}$: product-level future known inputs,
(4) $\mathbf{X}^{(l)} \in \mathbb{R}^{N\times R\times T\times d_l}$: product-region specific inputs, where $R$ is the total number of regions.

We also let $Y \in \mathbb{R}^{N\times T\times 1}$ denote the regional demand, that is, $Y \subset \mathbf{X}^{(l)}$.

Given a look-back window with length $C \geq 0$, and future-horizon $h$, we wish to generate the forecast

$$\widehat{Y}_{t,t+1:t+h} = f\left(Y_{t-C:t}, \mathbf{X}^{(g)}_{t-C:t}, \mathbf{X}^{(f)}_{t-C:t}, \mathbf{X}^{(g)}_{t-C:t}, \mathbf{X}^{(s)}; \boldsymbol{\theta}\right), \quad (1)$$

where $\boldsymbol{\theta}$ represents a collection of learnable parameters, $Z_{t-C:t} = (Z_{t-C}, \cdots, Z_t)$ for any variable $Z$ and $\hat{Y}_{t,t+1:t+h} = (\hat{Y}_{t,t+1}, \cdots \hat{Y}_{t,t+h})$ with $\hat{Y}_{t,t'}$ represents our forecast of the demand for week $t'$ made at week $t$. After the model $f$ is chosen, the parameters are tuned to optimize the aggregated loss through training

$$\text{Loss}(\boldsymbol{\theta}) = \sum_i \sum_r \sum_t \sum_h \ell(y_{i,r,t,h}, \widehat{y}_{i,r,t,h}), \quad (2)$$

where each forecast $\hat{y}_{i,r,t,h}$ is made at (product, region, forecast date, horizon) level.

To capture the uncertainty of demand, we decide to use "quantile loss" as the optimization objective: for the forecast $f$ at the $q$-th quantile with respect to the true demand $d$, the quantile loss $L$ (QL) is defined as

$$L_q(d, f) = q(d - f)_+ + (1 - q)(f - d)_+, \quad (3)$$

where $(\cdot)_+ = \max(\cdot, 0)$.

The weighted P50 and P90 QL, i.e., QL with $q \in \{0.5, 0.9\}$ aggregated all dimensions (e.g., products, regions and forecasting horizons) and then normalized by the aggregated demand, are widely used to measure the quality of forecasts and have been used as the loss function of training and evaluation metric for many previous works on multi-horizon time series forecasting [6, 23, 27].

## 4 Model

In this section, we present `RSight`, a novel deep-learning architecture for forecasting regional level demand. `RSight` uses the state-of-art `MQTransformer` [6] as the base architecture, and incorporates region-enhanced convolutions in the encoder. For clarity, we provide an overview of the main components of `MQTransformer` in Section 4.1. In Section 4.2, we discuss the innovations of `RSight`, i.e., the region-enhanced convolutions, which effectively incorporate regional-level features.

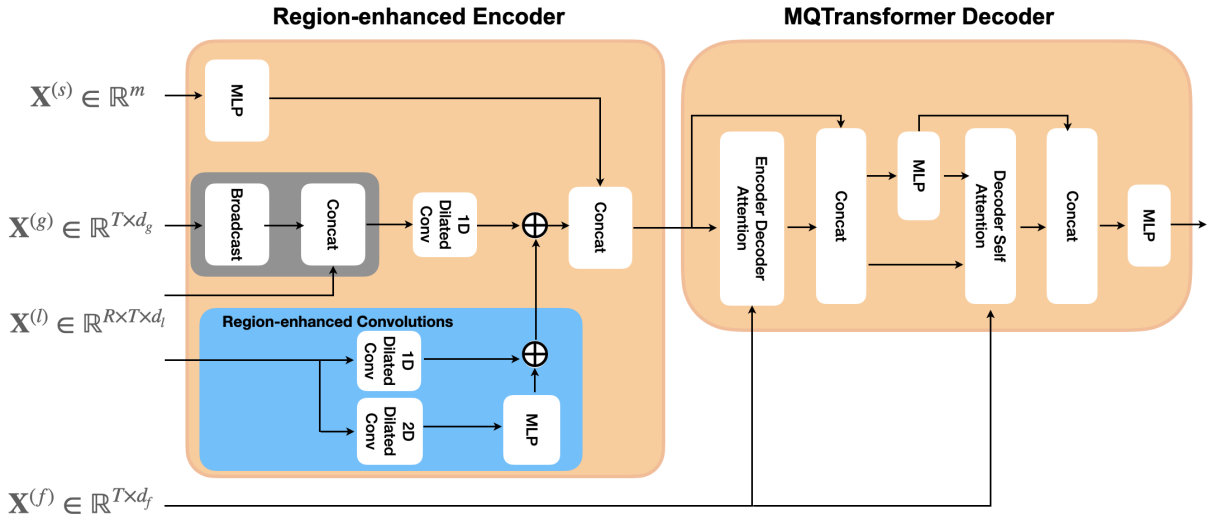



**Figure 1: Model architecture of `RSight`. We enhance the `MQTransformer` [6] encoder by combining the region-specific information $\mathbf{X}^{(l)}$ with region-agnostic information $\mathbf{X}^{(g)}$ early as well as adding region-enhanced convolutions highlighted in grey and blue correspondingly.**

## 4.1 MQTransformer

`MQTransformer` is built on two main architectural components: horizon-specific decoder-encoder attention and decoder self-attention. These components enable the model to (i) align different parts of the input history with each forecast horizon and (ii) incorporate information from past forecasts to reduce excess volatility.

The first module in `MQTransformer` produces a positional encoding for input features. Static features $\mathbf{X}^{(s)}$ and historical covariates $\mathbf{X}^{(g)}$ are passed into a positional encoder and concatenated:

$$
\begin{aligned}
r^{(s)} &= \mathrm{PE}^{(s)}\left(\mathbf{X}^{(s)}\right) \\
r^{(g)}_{t-C:t} &= \mathrm{PE}^{(g)}\left(\mathbf{X}^{(g)}\right) \\
r_{C-t,t} &= \left[\mathrm{Expand}\left(r^{(s)}\right), r^{(g)}_{t-C:t}\right],
\end{aligned}
$$

where $[x, y]$ denotes concatenation of two tensors. The positionally encoded vectors are then concatenated with demand features and passed to a Conv1D encoder:

$$
\tilde{h}_{t-C:t} = \mathrm{Conv1D}\left(Y_{t-C:t}, \mathbf{X}^{(g)}_{t-C:t}, \mathbf{X}^{(f)}_{t-C:t}, r_{t-C:t}\right) \tag{4}
$$

$$
h_{t-C:t} = \left[\tilde{h}_{t-C:t};\ \mathrm{Expand}\left(\mathrm{MLP}\left(\mathbf{X}^{(s)}\right)\right)\right]. \tag{5}
$$

To define the decoder, first let

$$
\mathrm{Attention}\left(Q, K, V; W\right) = \mathrm{SoftMax}\left(\frac{QW^Q(KW^K)^\perp}{\sqrt{d_{model}}}\right)VW^V.
$$

Then we define for each forecast horizon $l = 1, \ldots, H$ and snapshot $s, u = t - C, \ldots, t$:

$$
\begin{aligned}
c^{HS,l}_{s,u} &= \mathrm{Attention}\left([h_s; r_s, r_{s+l}], [h_u, r_u], h_s\right) \\
c^{\mathrm{agg}}_t &= \mathrm{MLP}([h_t; r_t]) \\
c_{s,r} &= [c_{t,1}; \cdots; c_{t,H} : c^{agg}_t] \\
c^{DS}_{t,h} &= \mathrm{Attention}(\{c_{s,h}, h_s, r_s\}_{s<t}),
\end{aligned}
$$

These are concatenated and passed into a shared output head:

$$
\hat{Y}_{t,t+h} = \mathrm{MLP}\left([c^{\mathrm{agg}}_t;\ c^{\mathrm{HS}}_{t,h};\ c^{DS}_{t,h};\ r_{t+h}]\right),
$$

which outputs $Q$ quantiles for the target horizon $h$.

## 4.2 RSight

Here, we introduce the model architecture of `RSight`, illustrated in Figure 1. Note that `MQTransformer` was designed for time-series data of a particular spatial granularity and does not leverage cross-regional information, limiting its effectiveness on regional forecasting tasks (as demonstrated in Section 5). Additionally, when forecasting regional demand, national-level demand can provide valuable contextual information about base regional demand trends at the regional level, suggesting that an effective model should incorporate both national and regional demand histories.

To address these limitations, `RSight` introduces an enhanced encoder layer designed to answer two fundamental questions:

(1) How can we effectively integrate region-specific features with region-agnostic (national-level) information?
(2) How can we leverage cross-regional dependencies to improve forecasting accuracy?

To address the first question, we propose a simple method for mixing the national-level features $\mathbf{X}^{(g)}_t$ with regional-specific features $\mathbf{X}^{(l)}_t$. In particular, we compute encoder input embeddings $U_t$

as follows:

$$U_t = \text{Concat}\left(\text{Broadcast}(\mathbf{X}_t^{(g)}), \mathbf{X}_t^{(l)}\right). \tag{6}$$

These embeddings $U_t$ are then passed to the Conv1D encoder defined in (4). To summarize, in RSight, instead of (4) that only considers product level historical covariates, we now have

$$\tilde{h}_{t-C:t} = \text{Conv1D}\left([Y_{t-C:t}, U_{t-C:t}, r_{t-C:t}]\right), \tag{7}$$

which also takes product-region level features.

To address the second question above, we introduce the region-enhanced conovolutions, which consists of an 1D and 2D dilated convolutions (illustrated in Figure 2). Formally, the region-enhanced convolutions do the following

$$V_t^{(1)} = \text{Conv1D}\left(X_{t-h:t}^{(l)}\right),$$
$$V_t^{(2)} = \text{Conv2D}\left(X_{t-h:t}^{(l)}\right),$$
$$H_t = V_t^{(1)} + \text{MLP}\left(V_t^{(2)}\right).$$

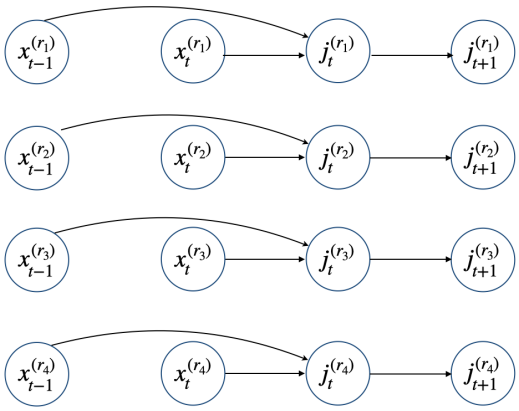
Intuitively, Conv1D layer provides region-specific time-series embeddings, while the Conv2D layer learns cross-region correlations. One motivation behind the use Conv2D for cross-region learning rather than, say, an MLP is that Conv2D can learn local regional patterns which are common across multiple regions. An important caveat of using Conv2D is a fixed, meaningful ordering of regions is required. An meaningless ordering such as randomly shuffling regions in the input tensor could make it harder for the model to learn related demand patterns. For example, if Miami, Orlando, and Tampa are placed far apart in the input representation, the model might struggle to efficiently capture the similar seasonal patterns in swimming pool supplies across these Florida cities, patterns that are driven by their shared climate and lifestyle characteristics. In our study, the natural ordering of US Zip2 provides implicit information of the geographical locations, i.e., Zip2 regions that are close in ordering are generally geographically close as well. Hence, we believe Conv2d can indeed learn meaningful cross-region information. The above intuition also points out interesting future direction to apply attention based learning across regions utilizing more region-specific information such as total population, income level, or other demographic characteristics.

The output of the region-enhanced convolution is then added back to the $\tilde{h}_{t-C}$ from (7) before feeding into the decoder.

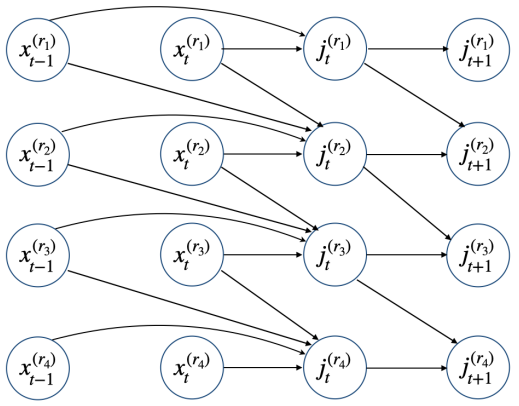
In summary, rather than (5) in MQTransformer, we now have

$$h_{t-C:t} = \left[\tilde{h}_{t-C:t} + H_{t-C:t};\ \text{Expand}\left(\text{MLP}\left(\mathbf{X}^{(s)}\right)\right)\right]$$

in RSight.



**(a) 1D dilated convolution layer: historical inputs only from own region is used**



**(b) 2D dilated convolution layer: the input from nearby regions are also used**

**Figure 2: Illustration of 2D dilated causal and 1D dilated causal layers**

## 5 Results

To benchmark RSight, we consider the following experimental setups. First, we compare RSight to a selection of state-of-the-art deep learning time-series forecasting models, including PatchTST [17], ChronosBolt [1], NHITS [2], NBEATS [20], and TS-Mixer [3], as well as MQTransformer [6]. We demonstrate that RSight outperforms all models on a large-scale regional demand forecasting problem.

Then, we perform a series of ablation studies. To begin with, we examine the scaling ability of RSight by increasing the number of series processed by the model. We demonstrate that RSight exhibits scaling laws which show promise for RSight as a large-scale time-series foundation model. We then ablate RSight on different activation functions to study whether the choice of activation functions affect the performance. Next, we experiment Attention and MLP based cross-region learning mechanisms instead of the base convolutional design. Finally, we demonstrate the importance of natural US Zip2 region ordering to RSight performance.

### 5.1 Large-Scale Regional Demand Forecasting

In this experiment, we consider a dataset of weekly sales for 15 million products sold from a large e-commerce retailer and includes time series features such as demand, promotions, holidays and detail page views as well as static metadata features such as catalog information. For each product, demand is observed in 100 Zip2 regions (defined as the first 2 digits of US zipcodes).

The dataset is divided into 10 million products whose data ranges from 2018 to 2022 and 5 million products whose data span 2023. At

the end of each week, we predict P50 and P90 quantiles for next 5 weeks.

In Table 1, we report the performance of RSight relative to the state-of-the-art benchmarks using Weighted Quantile Loss (WQL) as our performance metric. For products $i = 1, \ldots, N$ with observed demand $d_{i,t}$ and forecasts $f_{i,t}$ for $t = 1, ..., H$, WQL is defined

$$WQL(q) := \frac{\sum_i^N \sum_{t=1}^H L_q(d_{i,t}, f_{i,t})}{\sum_i^N \sum_{t=1}^H d_{i,t}}. \quad (8)$$

We use WQL as this metric gives greater weight to products with higher demand, which better aligns with business objectives since errors in high-demand products typically have a larger impact on revenue and customer satisfaction than errors in low-demand products.

We see that RSight outperforms recent developed time series deep learning models such as PatchTST and ChronosBolt by a large margin, with RSight improving P50 WQL by 200 bps (7%) and P90 WQL by 370 bps (13%). We propose two possible explanations for this performance improvement. First, each of the benchmark models do not leverage cross-region information, which is responsible for the performance improvement over MQTransformer. Second, the horizon-specific decoder-encoder attention and decoder self-attention inherited from the MQTransformer backbone process exogenous features efficiently. The significance of exogenous features for demand forecasting has been well-documented [30], and the efficient processing of such features is crucial for a performant forecasting model.

We also observe small improvement (11 bps in P50 and 20 bps in P90) from RSight against MQTransformer, showing the benefit of the region-enhanced convolutions. Due to the large scale of product population, improvements of such magnitude are still significant and entail sizable financial and operational impacts.

Details of the hyperparameter tuning for baseline models are reported in Appendix A.

**Table 1: WQL of RSight against state-of-art time series forecasting models**

| Model | P50 WQL | P90 WQL |
|---|---|---|
| RSight | **0.2748** | **0.2398** |
| MQTransformer | 0.2759 | 0.2418 |
| PatchTST | 0.2986 | 0.2773 |
| MLP | 0.5 | 0.9 |
| NHITS | 0.2983 | 0.2699 |
| ChronosBolt | 0.2948 | 0.2752 |
| TS-Mixer | 0.3764 | 0.3830 |
| NBEATS | 0.299 | 0.278 |

## 5.2 RSight scaling

We experiment with RSight and MQTransformer on different training data sizes to assess the scaling law on weighted quantile loss over the same inference data used in Table 1. To that end, we train both architectures on random samples of our full training data containing 10,000, 100,000, 1 million, and 10 million products respectively, across all US Zip2 regions.

Table 2 shows P50 and P90 weighted quantile loss for all cases. Both MQTransformer and RSight improve with the increase of training data, highlighting the returns to scaling. When we have small number of training data (less than 10,000 products), RSight do not outperform MQTransformer. One possible reason for that is that the number of data is too small for the region-enhanced convolution to learn useful information. With 1 million products, RSight start to outperform MQTransformer. While using 10 million products in training further improves the test error, the margin from RSight over MQTransformer is about the same.

**Table 2: WQL of RSight on different training data sizes**

| Training size | RSight P50 | RSight P90 | MQT P50 | MQT P90 |
|---|---|---|---|---|
| $10^4$ | 0.5001 | 0.8999 | 0.3101 | 0.2885 |
| $10^5$ | 0.3025 | 0.2651 | 0.2930 | 0.2713 |
| $10^6$ | 0.2802 | 0.2496 | 0.2828 | 0.2532 |
| $10^7$ | 0.2748 | 0.2398 | 0.2759 | 0.2418 |

## 5.3 RSight with different activation functions

Over time people have developed different activation functions for neural network. Here, we consider 5 activation functions for RSight and study their impact on WQL. Table 3 summarizes the result.

For P50, the choice of activation function does not lead to much difference (less than 7 bps across all models) while for P90 the deviations are larger (20 bps across different models) with Exponential Linear Unit (ELU) the best and PReLU the worst.

**Table 3: WQL of RSight with different activation function**

| Model | P50 WQL | P90 WQL |
|---|---|---|
| RSight-ReLU | **0.2748** | 0.2398 |
| RSight-LeakyReLU | 0.2755 | 0.2407 |
| RSight-PReLU | 0.2754 | 0.2409 |
| RSight-ELU | 0.2752 | **0.2389** |
| RSight-GELU | 0.2755 | 0.2408 |

## 5.4 RSight with different cross-region learning mechanisms

We evaluate two alternative cross-region learning mechanisms in place of the convolutional block in RSight. RSight-Attention applies multi-head self-attention along the region dimension at each time step and applies a demand-weighted distance matrix to mask our attention scores. In short, for any specific region, we will mask out regions with large distance to the target region. RSight-MLP uses a dimension-preserving linear transformation over the same axis. Table 4 compares these variants with the original RSight. Firstly, RSight-MLP produces higher WQL for both P50 and P90 compared to 2D convolution and attention based cross-region learning. This shows that the extra information from other regions cannot improve the forecast unless carefully utilized. Secondly, we observe RSight-Attention performs on par.

## 5.5 Region ordering

As mentioned in Section 4.2, the convolutional cross-region learning used by `RSight` is motivated by the intuition that the canonical ordering of US Zip2 regions provides a useful inductive bias. To verify this intuition, we randomly permute US Zip2 regions in our data and retrain `RSight` on the permuted data. The final row of Table 4 shows that disrupting natural region ordering leads to WQL degradation of 45 bps on P50 and 79 bps on P90, confirming our intuition.

**Table 4: WQL of `RSight` with different cross-region learning mechanisms and with shuffled region ordering**

| Model | P50 WQL | P90 WQL |
|---|---|---|
| RSight-Conv | **0.2748** | 0.2398 |
| RSight-Attention | 0.2756 | **0.2394** |
| RSight-MLP | 0.2793 | 0.2411 |
| RSight-Conv (shuffled regions) | 0.2793 | 0.2477 |

## 6 Conclusion

We introduced `RSight`, a novel deep neural network model designed to directly forecast product demand across predefined geographic regions within US. Our model uses `MQTransformer` as its backbone but can readily adapt to leverage other architectures such as SPADE [24], facilitating fast evaluation and deployment of state-of-the-art models across business applications. Our work demonstrates the feasibility of building a Deep Neural Network (DNN) model for regional demand forecasting using the same architectural backbone as national demand forecasting models. The next key research question is whether a single foundation model can effectively handle both national and regional forecasting simultaneously.

This work also shows several challenges to overcome along this path. First, we observe a noticeable impact of region-specific inputs on model performance, which requires new research on model's capability to handle multiple input granularity. Second, training across marketplaces requires the model to handle a flexible number of regions, a topic requiring effort beyond the scope of this paper. Finally, reconciling multiple tasks (e.g., national level forecasting vs region-level forecasting) inside a single DNN framework needs further study as well.

## References


[1] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. Expert Certification.

[2] Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. NHITS: Neural Hierarchical Interpolation for Time Series forecasting. In *The Association for the Advancement of Artificial Intelligence Conference 2023 (AAAI 2023)*, 2023.

[3] Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023.

[4] Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.

[5] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Carson Eisenach, Yagna Patel, and Dhruv Madeka. MQTransformer: Multi-Horizon Forecasts with Context Dependent and Feedback-Aware Attention. *Computing Research Repository.*, 8 2020.

[7] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

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

[10] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022.

[11] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Minbo Ma, Peng Xie, Fei Teng, Bin Wang, Shenggong Ji, Junbo Zhang, and Tianrui Li. Histgnn: Hierarchical spatio-temporal graph neural network for weather forecasting. *Information Sciences*, 648:119580, 2023.

[13] Yihong Ma, Patrick Gerard, Yijun Tian, Zhichun Guo, and Nitesh V Chawla. Hierarchical spatio-temporal graph neural networks for pandemic forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1481–1490, 2022.

[14] Fernando Moreno-Pino, Álvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough transformers: Lightweight and continuous time series modelling through signature patching. *Advances in Neural Information Processing Systems*, 37:106264–106294, 2024.

[15] Fernando Moreno-Pino, Pablo M Olmos, and Antonio Artés-Rodríguez. Deep autoregressive models with spectral attention. *Pattern Recognition*, 133:109014, 2023.

[16] Fernando Moreno-Pino and Stefan Zohren. Deepvol: Volatility forecasting from high-frequency data with dilated causal convolutions. *Quantitative Finance*, 24(8):1105–1127, 2024.

[17] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023.

[18] Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 2022.

[19] Kin G Olivares, O Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting*, 40(2):470–489, 2024.

[20] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

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

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017.

[23] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A Multi-horizon Quantile Recurrent Forecaster. In *31st Conference on Neural Information Processing Systems NIPS 2017, Time Series Workshop*, 2017.

[24] 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. ♠ spade ♠ split peak attention decomposition, 2024.

[25] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

[26] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with auto-correlation for long-term series forecasting. In M. Ranzato, A. Beygelzime, P.S. Liang, J.W. Vaughan, and Y. Dauphin, editors, *Advances in Neural Information Processing Systems 35 (NeurIPS 2021)*, 2021.

[27] Sitan Yang, Carson Eisenach, and Dhruv Madeka. Mqretnn: Multi-horizon time series forecasting with retrieval augmentation. 2022.
[28] Sitan Yang, Malcolm Wolff, Shankar Ramasubramanian, Vincent Quenneville-Belair, Ronak Mehta, and Michael Mahoney. Geann: Scalable graph augmentations for multi-horizon time series forecasting. In *KDD 2023 Workshop on Mining and Learning from Graphs*, 2023.
[29] Qingren Yao, Chao-Han Huck Yang, Renhe Jiang, Yuxuan Liang, Ming Jin, and Shirui Pan. Towards neural scaling laws for time series foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025.
[30] Muhammad Yasir, Yasmeen Ansari, Khalid Latif, Haider Maqsood, Adnan Habib, Jihoon Moon, and Seungmin Rho. Machine learning–assisted efficient demand forecasting using endogenous and exogenous indicators for the textile industry. *International Journal of Logistics Research and Applications*, 27(12):2867–2886, 2024.
[31] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.
[32] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *The Association for the Advancement of Artificial Intelligence Conference 2021 (AAAI 2021).*, abs/2012.07436, 2020.
[33] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.
[34] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

# A Baseline Hyperparameters

In order to select the hyperparameters for the baseline models, we took a random subset of 10000 products during the training window as a calibration set. We used 7000 products for training and 3000 products as validation and conducted grid search over possible parameters. Models with the best quantile loss on the validation were selected. Search spaces and final hyperparameters can be found in Table 5.

**Table 5: Hyperparameter search spaces for all models. Values in bold indicate the final selected configuration for each model.**

| Model | Parameter | Search Space |
|---|---|---|
| PatchTST | hidden_size | {64, **128**} |
| | n_heads | {4, **16**} |
| | encoder_layers | {**2**, 4} |
| | patch_len | {**8**, 16} |
| | d_model | {**16**, 32, 64} |
| | input_size | {**52**, 104} |
| | learning_rate | {0.0001, **0.001**} |
| | batch_size | {**32**} |
| NBEATS | num_layers | {**2**, 3} |
| | mlp_width | {128, **256**} |
| | n_harmonics | {1, **2**, 3} |
| | n_polynomials | {1, **2**, 3} |
| | d_model | {32, 64, **128**} |
| | input_size | {52, **104**} |
| | batch_size | {**32**} |
| | learning_rate | {0.0001, **0.001**} |
| NHITS | n_pool_kernel_size | {[2,2,1], **[4,2,1]**, [8,4,1], [16,8,1]} |
| | n_freq_downsample | {[168,24,1], [60,8,1],**[16,8,1]**, [1,1,1]} |
| | input_size_multiplier | {1, 3, 5} |
| | scaler_type | {None, "**robust**", "standard"} |
| | d_model | {32, **64**} |
| | input_size | {52, **104**} |
| | batch_size | {**32**} |
| | learning_rate | {0.0001, **0.001**} |
| ChronosBolt | input_patch_size | {**8**, 16} |
| | d_ff | {**512**, 1024} |
| | d_kv | {32, **64**} |
| | d_model | {32, **64**} |
| | input_size | {52, **104**} |
| | batch_size | {8, **16**} |
| | learning_rate | {0.0001, **0.001**} |
| TSMixer | n_block | {2, **3**} |
| | ff_dim | {16, 64, **128**, 256} |
| | residual_connection | {"add", "**cat**"} |
| | revin | {False, **True**} |
| | d_model | {16, 32, 64, **128**} |
| | input_size | {52, **104**} |
| | batch_size | {**32**} |
| | learning_rate | {0.0001, **0.001**} |
| MLP | num_layers | {**2**, 3, 4} |
| | hidden_size | {128, 256, **1024**, 2048} |
| | use_exog | {False, **True**} |
| | d_model | {**32**, 64} |
| | input_size | {**52**, 104} |
| | batch_size | {**32**} |
| | learning_rate | {0.0001, **0.001**} |