♠ SPADE ♠
Split Peak Attention DEcomposition

Malcolm Wolff *
Amazon
wolfmalc@amazon.com

Kin G. Olivares*
Amazon
kigutie@amazon.com

Boris Oreshkin
Amazon
oreshkin@amazon.com

Sunny Ruan
Amazon
jruan@amazon.com

Sitan Yang
Amazon
sitanyan@amazon.com

Abhinav Katoch
Amazon
abkatoch@amazon.com

Shankar Ramasubramanian
Amazon
sramasub@amazon.com

Youxin Zhang
Amazon
youxz@amazon.com

Michael W. Mahoney
Amazon
zmahmich@amazon.com

Dmitry Efimov
Amazon
defimov@amazon.com

Vincent Quenneville-Bélair
Amazon
quennv@amazon.com

Abstract

Demand forecasting faces challenges induced by Peak Events (PEs) corresponding to special periods such as promotions and holidays. Peak events create significant spikes in demand followed by demand ramp down periods. Neural networks like MQCNN [12, 6] and MQT [1] overreact to demand peaks by carrying over the elevated PE demand into subsequent Post-Peak-Event (PPE) periods, resulting in significantly over-biased forecasts. To tackle this challenge, we introduce a neural forecasting model called Split Peak Attention DEcomposition, SPADE. This model reduces the impact of PEs on subsequent forecasts by modeling forecasting as consisting of two separate tasks: one for PEs; and the other for the rest. Its architecture then uses masked convolution filters and a specialized Peak Attention module. We show SPADE's performance on a worldwide retail dataset with hundreds of millions of products. Our results reveal a reduction in PPE degradation by 4.5% and an improvement in PE accuracy by 3.9%, relative to current production models.

1 Introduction

Forecasting methods based on neural networks have produced accuracy improvements across multiple forecasting application domains such as large e-commerce retail [12, 1, 6, 8], financial trading [9], planning and transportation [5], and forecasting competitions [10, 7]. For large e-commerce, neural networks such as MQCNN [12, 6] and MQT [1] forecast product demand, based on product demand history, product attributes and known future information (e.g., promotions, planned sales, holidays), etc. Product demand frequently exhibits peaks caused by the impact of Peak Events (PEs) such as

* These authors contributed equally.

Preprint. Under review.

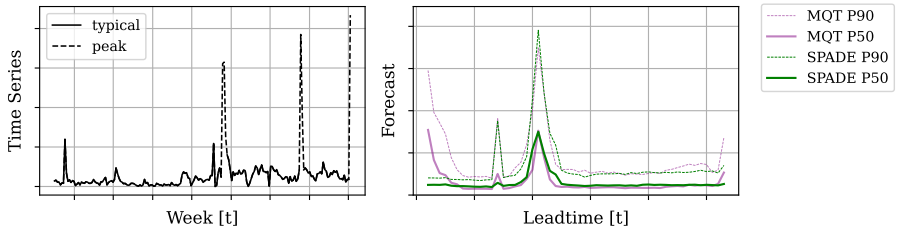

Figure 1: Illustration of "carry-over" degradation, visible in MQT's forecast downward trend. Peak values carry-over, degrading MQT's forecast accuracy, whereas SPADE does not exhibit such an effect.

promotions, deals, or holiday sales; forecast is particularly important during and after these peaks since we are more likely to run into inventory constraints.

The *carry-over* effect manifests itself as significant over-bias in the forecasts following demand spikes characteristic of PEs. Forecasting errors during and after PEs lead to logistical challenges: increased storage and operational costs, and often necessitate manual interventions to adjust forecasts, compromising the link between our provided forecast and downstream decisions. To handle this effect, filtering techniques such as [4, 2] have been used in traditional time series analysis. In this paper, we close the gap between such techniques and neural network based forecasts by developing a new forecasting architecture called SPADE capable of maintaining accuracy in the presence of PEs.

Our key contributions are summarized below:

(i) **Masked Convolutions.** We decompose the historical time series features into peak and non-peak components, based on *a priori* causal indicators. The convolutional encoder experiences only the historical demand without peaks.

(ii) **Peak Attention.** The peaks are encoded by a specialized attention mechanism we call Peak Attention. This module leverages *a priori* causal information, allowing the forecast to react quickly during and after PEs.

(iii) **Accuracy Improvements.** We expand upon the experiments' dataset size of MQCNN and MQT in [12, 1] from millions to hundreds of millions of series, and show post PE accuracy improvements of 4.5%, and PE accuracy improvements of 3.9% over MQCNN and MQT respectively.

We organize the rest of the paper as follows. We introduce our proposed SPADE method in Section 2. Section 3 contains main experiments and ablation studies. We summarize the results and mention future research directions in Section 4.

2 SPADE

Accurate forecasting requires integrating multiple inputs from other time series and data modalities, such as static information or exogenous factors, rather than relying solely on historical patterns. Additionally, effective models must jointly optimize across multiple tasks, considering not just standard metrics like MAE (Mean Absolute Error), but also the broader distribution of forecasts, or across different time granularities. Finally, robust forecasting must reliably capture both long-term trends and sharp peak events to be effective in practical settings.

SPADE is a Sequence to Sequence forecasting architecture [11] initially built for the purpose of product demand forecasting which integrates past time series data $\mathbf{x}_{[t]}^{(p)}$, static information $\mathbf{x}^{(s)}$ (e.g., product text features), and known future information $\mathbf{x}_{[t][h]}^{(f)}$ (e.g., world event indicators, countries and seller promotions). The model simultaneously predicts outcomes across multiple quantiles and levels of aggregation—each with unique patterns and noise characteristics—and leverages *a priori* event information to split Peak Events (PEs) from the remainder of the time series, eliminating "carry-over" effects many current time series forecasting models suffer from (see Figure 1).

Figure 2 displays the architecture. The PeakMask generates PE indicators from known future information, creating a mask to decompose the time series into PEs and non-PEs. The RobustConvolution

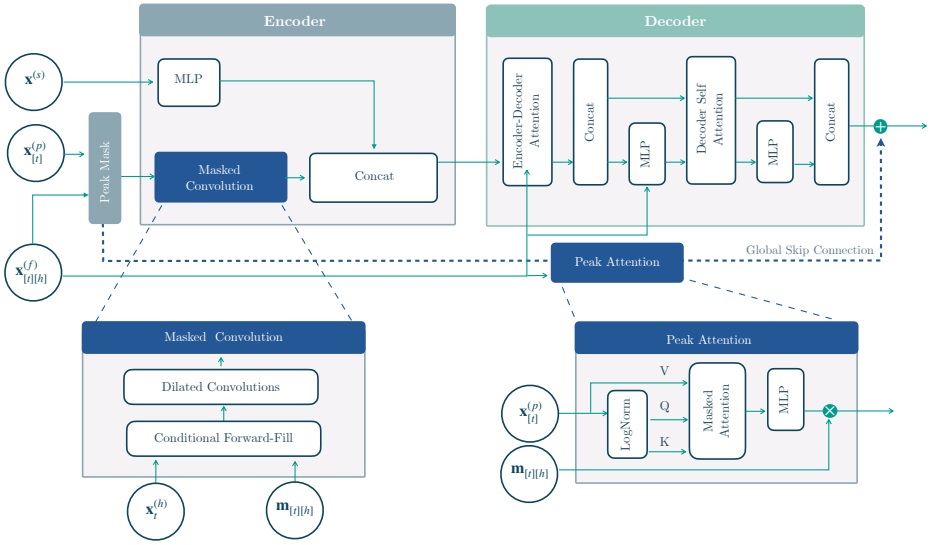

Figure 2: SPADE decomposes its temporal features to distinguish usual behavior from peaks.

Table 1: Empirical evaluation of the weighted quantile loss (WQL). We evaluate P50 and P90 WQL overall, during and after PEs. The best result is shown in bold, lower is better.

Metric	SPADE	MQCNN	MQT
P50 WQL	**0.9912**	1.1842	1.0000
P90 WQL	**0.9935**	1.1990	1.0000
P50 WQL_{PE}	**0.9672**	0.9740	1.0000
P90 WQL_{PE}	**0.9557**	1.0068	1.0000
P50 WQL_{PPE}	0.9576	**0.9386**	1.0000
P90 WQL_{PPE}	0.9520	**0.9380**	1.0000

block filters PEs from the historic inputs with a forward-fill operation and encodes the result with a series of dilated convolutions. The PeakAttention block forecasts PE magnitudes using past PE information and known future information, and the PE and non-PE partial predictions are summed to produce the final forecast. Architecture details are provided in Appendix A.

Forecast accuracy is evaluated with the weighted quantile loss (WQL)

$$\text{WQL}(\mathbf{y}, \hat{\mathbf{y}}^{(q)}; q, \mathcal{I}, \mathcal{H}) = \frac{\sum_i \sum_t \sum_h \text{QL}\left(y_{i,t,h},\ \hat{y}^{(q)}_{i,t,h}(\boldsymbol{\theta}); q\right)}{\sum_i \sum_t \sum_h y_{i,t,h}}, \tag{1}$$

where $\text{QL}\left(y,\ \hat{y}^{(q)}; q\right) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+$ is the quantile loss function, $\hat{y}^{(q)}$ denotes the estimated quantile, $\boldsymbol{\theta}$ denotes a model in the class of models Θ defined by the model architecture. We optimize $\boldsymbol{\theta}$ by minimizing the numerator of equation (1) summed across the quantiles of interest. See Appendix B for details. In addition to WQL, we evaluate the WQL during PEs and Post-PE (denoted PPE) to capture accuracy at PEs and the "carry-over" effect, respectively; letting $\mathcal{I}_{\text{PE}} \equiv \{i \in \mathcal{I} \mid d_{i,t} = 1\}$ where $d_{i,t}$ represents time series deal information, we have

$$\text{WQL}_{\text{PE}} = \text{WQL}(\mathbf{y}, \hat{\mathbf{y}}^{(q)}; q, \mathcal{I}_{\text{PE}}, \mathcal{H}_{\text{PE}}) \quad \text{and} \quad \text{WQL}_{\text{PPE}} = \text{WQL}(\mathbf{y}, \hat{\mathbf{y}}^{(q)}; q, \mathcal{I}, \mathcal{H}_{\text{PPE}}) \tag{2}$$

where $\mathcal{H}_{\text{PE}}$ are the horizons including a PE occur, and $\mathcal{H}_{\text{PPE}}$ are the horizons after. A typical demand pattern indicating seasonality, PE demand peak and PPE carry-over effect is shown in Figure 1.

3 Experiments

The dataset consists of hundreds of millions of series across several countries covering three years training data (2019-2022) and one year for evaluation (2023). Table 1 displays results of empirical

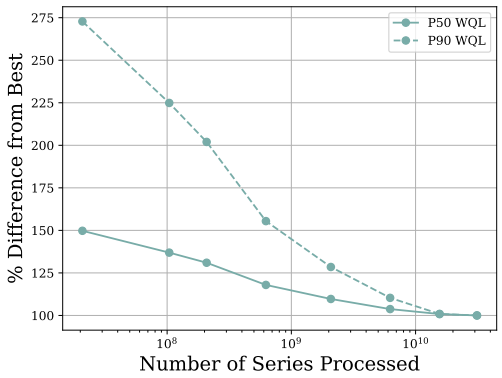

Figure 3: SPADE shows evidence of forecast accuracy scaling with training time series.

evaluation for SPADE, MQCNN and MQT across countries. MQCNN and MQT show a distinct trade-off surrounding PEs. For each country, the carry-over behavior of MQCNN's encoder under-biases PE forecasts, resulting in an inflated WQL during these events, but low PPE error. This results in a 0.6% degradation in P50 WQL and 5% degradation in P90 WQL relative to SPADE during PEs. We note, however, that due to MQCNN's inability to capture peaks, the post-peak accuracy remains high. We find a 2% improvement in P50 PE WQL and 2% improvement in P90 WQL relative to SPADE post peak.

On the other hand, MQT's attention mechanisms do lift demand forecasts during PEs to improve accuracy during events, but due to the carry-over of the time series embeddings, results in PPE degradation. This results in a 4% degradation in P50 WQL and 5% degradation in P90 WQL in relative to SPADE post-peak. Moreover, while MQT does exhibit higher accuracy during PEs than MQCNN, SPADE's explicit attention to these events result in large improvement. SPADE outperforms MQT by 3% on P50 WQL and 4% on P90 WQL during peaks.

To further clarify and analyze the source of improvements we performed an ablation study on variants of the MQCNN and MQT, for a smaller dataset in Appendix C.

Finally, we study the scaling properties of our model. Our results, represented by Figure 3, show that forecast accuracy scales with the number of series during training, with P90 WQL improving more substantially than P50 WQL.

4 Conclusions

We introduced a novel forecasting method, SPADE, that combines two complementary techniques (splitting temporal features into peak and non-peak components before encoding them, and a specialized peak attention module to enhance forecast sharpness) to identify and deal with spikes in realistic forecasting systems. SPADE alleviates the peak *carry-over* effect, showing higher accuracy during and after the peaks events. In our retail demand forecasting experiments, we observe a PE accuracy improvement of 3.9%, and a reduction of the post peak degradation of 4.5%, when compared to MQCNN and MQT production models.

In our empirical evaluations, we applied peak masking convolutions to variants of MQCNN and MQT, but our SPADE method is not restricted to convolutional encoders. Exploring its application to other recurrent encoders such as LSTMs, attention mechanisms, or MLPs presents a promising avenue for future research. Moreover, the current implementation of SPADE relies on pre-existing causal and PE indicators, but there is potential to enhance the methodology by incorporating unsupervised detection techniques to identify PEs autonomously.

References

[1] Kevin C. Chen, Lee Dicker, Carson Eisenach, and Dhruv Madeka. MQTransformer: Multi-Horizon Forecasts with Context Dependent Attention and Optimal Bregman Volatility. In Maria Florina Balcan and Marina Meila, editors, *In: Proceedings of 8th SIGKDD International Workshop on Mining and Learning From Time Series - Deep Forecasting: Models,*

Interpretability, and Applications (KDD' 22). ACM. Association for Computing Machinery, 8 2022.

[2] Ruben Crevits and Christophe Croux. Forecasting with robust exponential smoothing with damped trend and seasonal components. *Journal of Statistical Software, Articles*, 11 2016.

[3] Diederik P. Kingma and Jimmy Ba. ADAM: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations (ICLR), San Diego, 2015.

[4] Nora Muler, Daniel Peña, and Víctor J. Yohai. Robust estimation for ARMA models. *The Annals of Statistics*, 37(2):816 – 840, 2009.

[5] Jiafan Yu Nikolay Laptev and Ram Rajagopal. Reconstruction and regression loss for time-series transfer learning. *Proceedings of the Special Interest Group of Knowledge Discovery and Data Mining (SIGKDD)*, 2018.

[6] Kin G. Olivares, Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting, accepted*, Preprint version available at arXiv:2110.13179, 2023.

[7] 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.

[8] Vincent Quenneville-Belair, Malcolm Wolff, Brady Willhelme, Dhruv Madeka, and Dean Foster. Distribution-free multi-horizon forecasting and vending system. In *KDD 2023 International Workshop on Mining and Learning from Time Series (MileTS)*, 2023.

[9] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

[10] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 07 2019.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[12] 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.

[13] Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

A Architecture Details

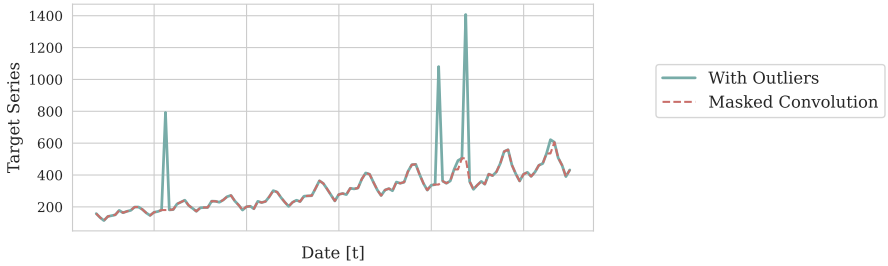

Figure 4: Masked convolutions enhance neural forecasting architectures by filtering peaks before inputting temporal features to the encoder, thus mitigating the peak carry-over effect.

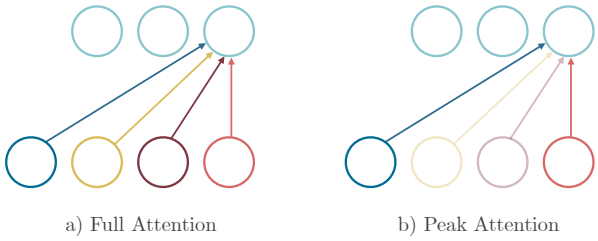

Figure 5: The Peak Attention module regularizes the classic attention mechanism by sparsifying its weights using future covariate information.

Masked Convolution. We use the causal indicators from the known future information $\mathbf{x}^{(f)}_{[t][h]}$, to create a mask that identifies peak temporal observations $\mathbf{m}_{[t][h]}$. The peak mask $\mathbf{m}_{[t][h]}$ is then used to decompose the time series into peak and non-peak observations. The $\mathbf{x}^{(p)}_{[t]}$ inputs of the convolutional encoder are thus filtered of their peak observations following:

$$\tilde{\mathbf{x}}^{(p)}_{t} = \begin{cases} \mathbf{x}^{(p)}_{t} & \text{if} \quad \mathbf{m}_{t} = 0 \\ \mathbf{x}^{(p)}_{t^{*}} & \text{if} \quad \mathbf{m}_{t} = 1, \; t^{*} = \min_{\tau \leq t} \mathbf{m}_{\tau} = 0 \end{cases} \tag{3}$$

$$\mathbf{e}^{(p)}_{[t]} = \text{Convolution}(\tilde{\mathbf{x}}^{(p)}_{[t]}). \tag{4}$$

Peak Attention. This module computes a forecast update $\Delta_{[t][h]}$ using the peak mask $\mathbf{m}_{[t][h]}$, the past information $\mathbf{x}^{(p)}_{[t]}$ and known future information $\mathbf{x}^{(f)}_{[t][h]}$ to compute the following operations:

$$\mathbf{q}_{[t][h]} = \text{MLP}(\mathbf{e}^{(p)}_{[t]}, \mathbf{x}^{(f)}_{[t][h]}) \quad \mathbf{k}_{[t][h]} = \text{MLP}(\mathbf{e}^{(p)}_{[t]}, \mathbf{x}^{(f)}_{[t][h]}) \quad \mathbf{v}_{[t][h]} = \text{MLP}(\mathbf{e}^{(p)}_{[t]}, \mathbf{x}^{(f)}_{[t][h]}) \tag{5}$$

$$\mathbf{H}_{[t][h]} = \text{SoftMax}\left(\mathbf{q}_{[t][h]} \times \mathbf{k}^{\intercal}_{[t][h]} \times \mathbf{m}_{[t][h]}\right) \times \mathbf{v}_{[t][h]}, \tag{6}$$

$$\Delta_{[t][h]} = \text{MLP}(\mathbf{H}_{[t]}) \times \mathbf{m}_{[t][h]}, \tag{7}$$

where $\mathbf{q}_{[t][h]}$, $\mathbf{k}_{[t][h]}$, and $\mathbf{v}_{[t][h]}$ are horizon-specific query, key, and value tensors.

The attention weights are masked to only use PEs from the past; in contrast to a complete horizon-time attention module, `PeakAttention` requires only a fraction of the computation, as it only searches historical PEs instead of the full history to induce the forecasts' sharpness.

$$\hat{\mathbf{y}}_{[t][h]} = \Delta_{[t][h]} + \text{MLPDecoder}\left(\mathbf{e}^{(p)}_{[t]}, \mathbf{e}^{(s)}\right), \tag{8}$$

where $\mathbf{e}^{(p)}_{[t]}$ are the encoded masked historic temporal features and $\mathbf{e}^{(s)}$ are the encoded static features. We incorporate a global skip connection to adjust the predictions of the multi-horizon MLP decoder, effectively decomposing the forecasts into PEs and a baseline.

B Training Methodology and Hyperparameter Selection

Table 2: The SPADE architecture parameters configured once. The second panel controls the optimized parameters, we only considered learning rates and training epochs. (*The effective SGD batch is multiplied by the number of GPUs in the execution cluster.)

PARAMETER	Notation	Considered Values		
		SPADE	MQT	MQCNN
Single GPU SGD Batch Size*	-	32 (10,240)	32 (10,240)	32 (10,240)
Main Activation Function	-	ReLU	ReLU	ReLU
Max Temporal Convolution Kernel Size	-	32	32	32
Temporal Convolution Layers	-	6	6	6
Temporal Convolution Filters	-	30	30	30
Static Encoder D.Multip. $(\alpha \times \lvert\sqrt{x}^{(s)}\rvert)$	-	30	30	30
Future Encoder Dimension (hf1)	-	50	50	50
Horizon Agnostic Decoder Dimensions	-	100	100	100
Horizon Specific Decoder Dimensions	-	20	20	20
PeakAttention Number of Heads	-	4	4	4
Learning Rates	-	$\{0.001, 0.0001\}$	$\{0.001, 0.0001\}$	$\{0.001, 0.0001\}$
Number of Epochs	-	$\{10, 20, 30\}$	$\{10, 20, 30\}$	$\{10, 20, 30\}$

Training Methodology. Let $\boldsymbol{\theta}$ be a model that resides in the class of models Θ defined by the model architecture. Let $\mathcal{A}$ the dataset's products, and $\mathcal{H}$ the horizon defined by lead times and spans. We train a quantile regression model by minimizing the following multi-quantile loss:

$$\min_{\boldsymbol{\theta}} \sum_{q} \sum_{i} \sum_{t} \sum_{h} \mathrm{QL}\left(y_{i,t,h},\ \underline{\hat{y}}^{(q)}_{i,t,h}(\boldsymbol{\theta});\ q\right), \tag{9}$$

for products $a \in \mathcal{A}$, time t and horizon $h \in \mathcal{H}$, and $\underline{\hat{y}}^{(q)}$ denotes the estimated quantile[1]. We optimize SPADE using stochastic gradient descent with *Adaptive Moments* (ADAM; [3]). Details on the implementation and optimization methodology and variants are available in Appendix B.

Hyperparameter Selection. The cornerstone of the training methodology for SPADE and the MQT and MQCNN baseline models is the definition of the training, validation and test sets. For the worlwide retail dataset the training set consists of the first four years of observations before a year of validation data. Since SPADE is a production model the test set is incrementally updated as new data becomes available, the model selection is performed using an online learning approach.

For the hyperparameter selection, we only consider the exploration learning rates and then number of SGD. See Table 2 the configuration space along with particular hyperparameters for each dataset.

[1]During training, demand and forecasts are normalized by the length of the horizon h.

C SPADE Architecture Ablation Study

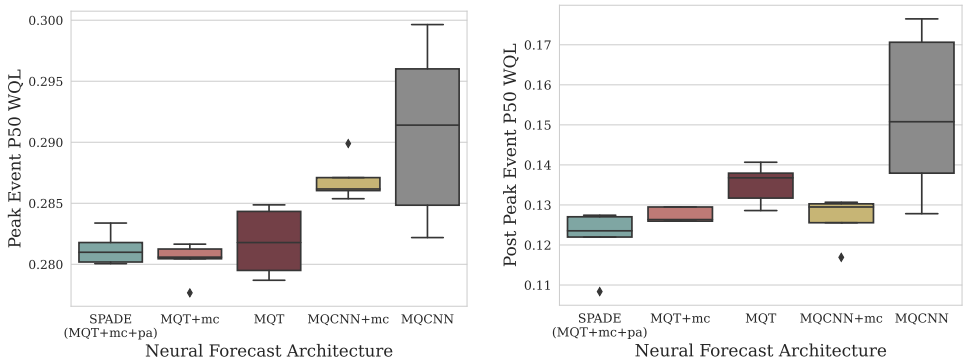

Figure 6: Peak Attention and Masked Convolutions enhance both MQCNN and MQT architectures by filtering peaks before inputting temporal features to the encoder, thus mitigating the peak carry-over effect. This approach significantly reduces training variance and enhances the forecast accuracy.

Table 3: Evaluation of P50 and P90 probabilistic forecasts averaged over 10 random seeds along their 95 percent confidence interval ($\pm$), lower is better. All hyperparameters were kept constant across all architecture variants. Average percentage difference relative to control in the first column.

		MQT				MQCNN		
		Diff	Peak Attention	Masked Conv	Original	Diff	Masked Conv	Original
Overall	P50 WQL	-1.237	0.0918±0.0041	0.0946±0.0107	0.0930±0.0068	2.333	0.0939±0.0034	0.0918±0.0057
	P90 WQL	-5.786	0.0629±0.0007	0.0646±0.0029	0.0668±0.0020	-1.757	0.0642±0.0005	0.0653±0.0007
Peak	P50 WQL	-0.197	0.2813±0.0024	0.2803±0.0027	0.2818±0.0049	-1.342	0.2869±0.0031	0.2908±0.0129
	P90 WQL	-4.193	0.3483±0.0064	0.3422±0.0096	0.3636±0.0200	-3.736	0.3545±0.0104	0.3683±0.0323
PostPeak	P50 WQL	-9.966	0.1217±0.0137	0.1274±0.0033	0.1351±0.0086	-17.121	0.1266±0.0101	0.1527±0.0364
	P90 WQL	-16.151	0.0762±0.0038	0.0777±0.0017	0.0909±0.0051	-17.834	0.0775±0.0036	0.0944±0.0111

We performed ablation studies on variants of the `MQCNN` and `MQT`. We attribute the effectiveness of `SPADE` in mitigating the peak *carry-over* effect to the incorporation of masked convolutions and temporal feature splitting. We simplified the experimental setup in this ablation study, using the `Tourism-L`dataset, a detailed Australian Tourism Dataset comes from the National Visitor Survey, managed by the Tourism Research Australia agency, it is composed of 555 monthly series from 1998 to 2016 organized geographically [13]. We introduce peaks in the time series, simulating a 3% data contamination using normal noise with the variance of each series to achieve better control in the experiments, making it easier to evaluate the accuracy of around PEs.

We consider the forecasting task, where we produce P50 and P90 quantile forecasts for the last twelve months of all `Tourism-L`series, that we evaluate using the Weighted Quantile Loss (WQL) defined in Equation 1, but in contrast to the main experiment we only consider weekly forecasts. We differentiate between overall and post PEs (PPEs), to focus on the *carry-over* effect. As shown in Table 3, the masked convolution filters predictions and consistently improves its original counterpart. Relative post PEs P50 improvements of 9.96% for `MQT` and 17.12% for `MQCNN`. Figure 6 and Table 3, comparing `MQCNN`/`MQT` with and without masked convolution reveal a notable enhancement in PPE accuracy when the convolution is included. In addition combining the masked convolutions with the `PeakAttention` module produces the best results, as components appear to complement each other.