Adjustable Context-aware Transformer

Sepideh Koohfar and Laura Dietz

University of New Hampshire, Durham NH, USA
{Sepideh.Koohfar, Laura.Dietz}@unh.edu

Abstract. We propose a multi-horizon forecasting approach that accurately models the underlying patterns on different time scales. Our approach is based on the transformer architecture, which across a wide range of domains, has demonstrated significant improvements over other architectures. Several approaches focus on integrating a temporal context into the query-key similarity of the attention mechanism of transformers to further improve their forecasting quality. In this paper, we provide several extensions to this line of work. We propose an adjustable context-aware attention that dynamically learns the ideal temporal context length for each forecasting time point. This allows the model to seamlessly switch between different time scales as needed, hence providing users with a better forecasting model. Furthermore, we exploit redundancies arising from incorporating the temporal context into the attention mechanism to improve runtime and space complexity. Our experiments on several real-world datasets demonstrate significant performance improvements over existing state-of-the-art methodologies. The code for reproducing the results is open sourced and available online[1].

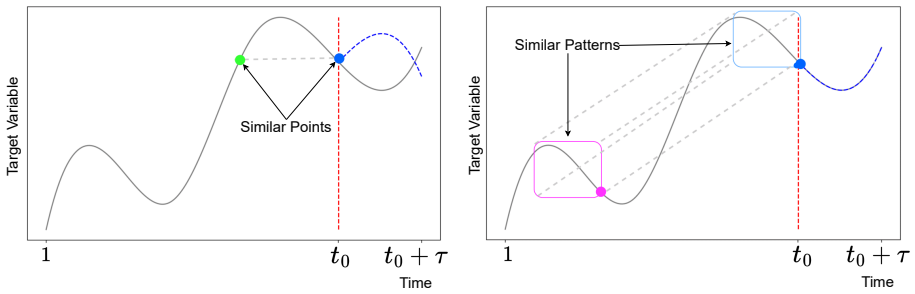
Keywords: Time Series Forecasting · Temporal Systems · Neural Networks

1 Introduction

Time series forecasting is an important problem across many domains, such as economics [4, 34], retail [29, 7], healthcare [19], and sensor network monitoring [23]. In such domains, users are interested in future forecasts based on the historical data to glean insights. Multi-horizon forecasting is a critical demand for many applications, such as early severe weather events forecasting and travel planning based on traffic congestion.

Recurrent Neural Networks (RNNs) have been applied to model repeating patterns in time series [24, 25], however RNNs and their variants are not able to leverage information from the longer past as needed to model long-term dependencies. Previous studies have demonstrated that RNNs including Long Short Term Memory networks (LSTMs) [12] might also fail to capture long-term patterns of dependency when information from the past is gradually overwritten by information from recent observations [15].

[1] https://github.com/SepKfr/Adjustable-context-aware-transfomer



a) Point-wise attention b) Temporal attention

Fig. 1. A synthetic example of predicting the future after the dashed red line given the preceding data. The attention mechanism would choose the points that are most similar to the query point at t_0 (depicted in blue) and predict a similar trajectory. In the case of (a) point-wise attention, this would be the green point, which would result in an erroneous forecast following the blue dashed line, where the correct forecast is depicted in black. In the case of (b) temporal attention the similarity is depending on the context, depicted as rectangle, and the pink rectangle is determined to exhibit most similar behavior. This results in an accurate forecast, depicted as blue dashed line.

Transformers [30] can incorporate any observations of the series (potentially skipping over non-relevant data points) which renders them more suitable for capturing similarities in the longer past. We hypothesize that these similarities are critical for achieving accurate forecasts. However, the basic attention mechanism in transformers estimates the similarity based on a point-wise vector of the query and key, each representing individual time steps, thereby ignoring the temporal context surrounding the query and key. As depicted in Figure 1 estimating similarities based on point-wise vectors without incorporating the temporal context might lead to misleading predictions, when observations have high point-wise similarities but exhibit different temporal behaviours (Figure 1 left).

Many approaches are based on the hypothesis that in a multi-layer model, the temporal context can be absorbed into the representation of the query and key from a previous layer. However, there are two shortcomings. 1) Previous layers suffer from the same lack of temporal understanding. 2) Such a mechanism operates indirectly which provides little insights for more explainability. We strive for a direct approach that is even effective as a single-layer model, as it obtains better performance while using fewer resources.

Li et al [18] address this shortcoming with a Convolutional Neural Network (CNN) transformer. CNNs are used as a preliminary layer to inform points with context information to feed into the transformer stack. However, our experimental results demonstrate that integrating a fixed length temporal context limits the degree of flexibility to detect similarities on different time scales in order to improve the forecasting quality.

We dive into these issues and investigate the importance of incorporating a flexible temporal context into the attention mechanism. Our contributions are summarized as follows:

We propose the *adjustable context-aware* attention, an attention mechanism that dynamically learns the optimal temporal context length for each query and key pair to obtain the best forecasting results. We propose the adjustable Context-aware Transformer (ACAT) which replaces the basic attention mechanism with our adjustable context-aware attention.
We increase the efficiency of our approach by exploiting redundancies in temporal contexts. While it seems counter intuitive to first introduce then remove redundancies, the result is a more efficient model overall.
We successfully apply our proposed architecture to real world time series datasets to validate its potential in generating more accurate predictions.

2 Problem Definition

Given the input data prior to time step t_0, the task is to predict the variables of interest for multiple steps into the future from t_0 to $t_0 + \tau$.

Given the previous time series observations of variables of interest $\boldsymbol{y}_{1:t_0} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{t_0}]$, and time series covariates that are known over both the historical and forecasting period $\boldsymbol{x}_{1:t_0+\tau} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t_0+\tau}]$, we predict the variables of interest for the next τ time steps $\boldsymbol{y}_{t_0+1:t_0+\tau} = [\boldsymbol{y}_{t_0+1}, \boldsymbol{y}_{t_0+2}, \ldots, \boldsymbol{y}_{t_0+\tau}]$. Where $\boldsymbol{y}_i \; 2 \; R^{d_y}$ and $\boldsymbol{x}_i \; 2 \; R^{d_x}$

For a univariate problem each $\boldsymbol{x}_i$ contains the information of static time-based covariates such as hour of the day and day of the week. However, we also include other exogenous covariates to $\boldsymbol{x}_i$, when dealing with a multivariate problem. In this paper we focus on generating univariate predictions where the total number of target variables at each step is one ($d_y = 1$), although the problem can be generalized to predict multiple target variables at a time step ($d_y > 1$).

3 Related Work

Time series forecasting methods can be categorized into classical and neural network methods. Prominent examples of classical methods include ARIMA [3] and state space models (SSMs) [8]. ARIMA and SSMs require the expertise of practitioners to manually select the autocorrelation structure, seasonality, trend, and other explanatory variables. Additionally, the core assumption of classical methods, such as stationarity and homoscedasticity of the time series data, make them unsuitable for sporadic and nonuniform large data.

Neural network methods have been widely applied to time series forecasting to model the non-linearity in large-scale data across related time series. Deep neural networks have been proposed to model the interacting patterns among time series and they have demonstrated strong performance improvements over

traditional time series models [24, 1, 22]. Many deep learning architectures depend on RNNs to model non-trivial time series data [24, 25, 32, 10]. Deep AR [25] generates parameters of a one-step-ahead Gaussian distribution using stacked layers of LSTM [13]. The Deep State Space Model (DSSM) [24] uses LSTMs to generate parameters of a linear state space model at each time step. Multi-horizon Quantile Recurrent Forecaster (MQRNN) [32] uses an RNN architecture as an encoder to generate a context vector for multi-layer perceptrons to generate multi-step forecasts.

Transformers have shown superior performance in modeling long-term dependencies compared to RNNs [18, 9]. The Attention mechanism has been applied to a variety of tasks such as translation [30], image classifications [31], music generation [14], and tabular learning [2]. Recently, the attention mechanism has gained popularity in the time series community as well [18, 6, 20, 21, 27]. To include temporal information in the query-key similarity of the attention mechanism, several works benefit from CNNs [17, 18, 26]. CNN-trans [18] uses a convolutional processing layer to integrate the temporal context into the attention mechanism to build an autoregressive transformer-based model. Temporal Convolution Attention-based Network (TCAN) [28] follows a similar approach by applying dilated convolution to the output of a temporal attention layer to model the short- and long-term temporal behavior. However the aforementioned approaches use a convolutional filter with a fixed size controlled by a hyperparameter. In natural language processing domain, Chen et al [5] demonstrate the usefulness of a max pooling layer on top of different CNN filters by proposing a Dynamic Multi-pooling Convolutional Neural Network (DMCNN). Albeit in a different domain, this idea is related in spirit to our approach.

Recently, new approaches have developed efficient transformer models to predict for long-time series forecasting problems. Informer [33] uses KL-divergence and ProbAttention to select the most prominent queries and reduces the complexity to $O(L \log L)$. Autoformer [11] includes the series decomposition as an inner block of transformers and replaces the attention mechanism with the auto-correlation among sub-series to achieve $O(L \log L)$ complexity. Our contribution is complementary to these extensions since our attention mechanism can serve as a substitute for auto-correlation and attention block in ProbAttention.

4 Methodology

In this section, we detail shortcomings to point-wise attention and elaborate on principles of context-based temporal attention. We develop a self-adjusting model that will select the most appropriate filter size for each forecasting decision. We remove redundancies of this model via a subsampling approach.

4.1 Background: Issues Arising from Point-wise Attention

Given time series data, a basic single-layer transformer model with masked scaled dot-product QKV attention would predict the layer output $\boldsymbol{y}_i$ at time step i as

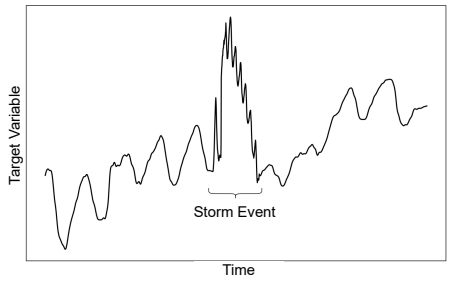


Fig. 2. An example of the temporal behaviour of the target variable taken from the watershed dataset. The center of the plot depicts a stormy period with higher dynamical behavior. Shorter context sizes are more appropriate for modeling temporal behavior in these periods.

$\boldsymbol{y}_i = \texttt{softmax}(\sum_j {}_i \boldsymbol{a}_{ij}\boldsymbol{v}_j)$ with attention $\boldsymbol{a}_{ij} = \texttt{softmax}\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_j / \sqrt{d}\right)$ where query, key, and value vectors are derived via $\boldsymbol{q}_i = \texttt{proj}(\boldsymbol{x}_i)$, $\boldsymbol{k}_j = \texttt{proj}(\boldsymbol{x}_j)$, and $\boldsymbol{v}_j = \texttt{proj}(\boldsymbol{x}_j)$ using three different multi-layer perceptron-style projections of inputs to vectors with dimension d.

We call this form of attention point-wise, because it does not incorporate the temporal context, as we are solely considering the information at time points i and j. Figure 1 illustrates a forecasting issue that might arise because of the point-wise attention mechanism in transformers. The canonical approach to incorporate the temporal context into the attention mechanism is to use a multi-layer model. The hope is that in a multi-layer architecture, the previous layers provide an appropriate representation of the temporal context. However, previous layers are also suffering from the problem induced by point-wise attention. Therefore, while multi-layer transformers can theoretically learn a good temporal representation, the architecture is an impediment to good performance. In the following we address this issue with a simple-to-implement approach to directly incorporate the temporal context and demonstrate in the experimental evaluation that even a single layer of our approach can outperform a multi-layer transformer model.

4.2 Temporal Attention

The first step towards a better model is to include the temporal context into the attention. This idea has been discussed under the name convolutional attention [18] or temporal attention [28].

This is achieved by deriving query and key vectors from the context of length L preceding the time step i. Denoting this temporal context $i_{<L} = [(i \quad (L \; 1)), \ldots, i]$, the only modification to the temporal attention is how query and key vectors are obtained: $\boldsymbol{q}_i = \texttt{proj}(\boldsymbol{x}_{i_{<L}})$ and $\boldsymbol{k}_j = \texttt{proj}(\boldsymbol{x}_{j_{<L}})$. The attention follows as: $\boldsymbol{a}_{ij} = \texttt{softmax}\left(\boldsymbol{q}_i^\intercal \boldsymbol{k}_j / \sqrt{d}\right)$.

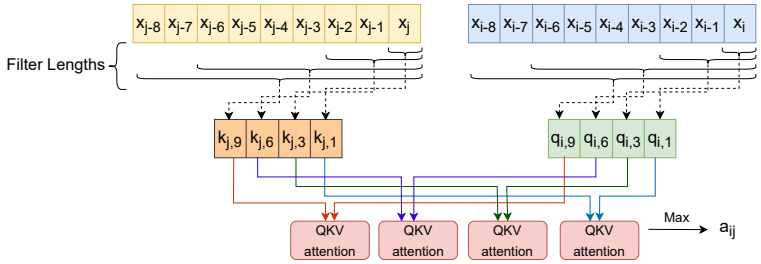


Fig. 3. An example of the adjustable context-aware attention architecture with context lengths $\mathcal{L} = \{1, 3, 6, 9\}$. We consider different context sizes to project the input data at time step i and j to create a set of context-aware query and key vectors (the projections is indicated in dashed arrows). Attention score a_{ij} is governed by the highest similarity to query at time step i and key at time step j.

Note that $\boldsymbol{x}_{j<L}$ 2 $\mathbb{R}^{d_x \quad L}$ is a matrix, which can be interpreted as being re-shaped into a vector.

While this kind of attention is considering the temporal context, the issue is that this context is of a fixed length L, which needs to be either pre-determined or tuned as a hyperparameter. The common belief is that even if the ideal context length L would be unknown, choosing a sufficiently large context length L L is sufficient. It is based on the hope that end-to-end training will set the parameters of the MLP-projection to ignore parts that are not needed.

However, we hypothesize that the noise introduced by excessively large context sizes will inhibit good performance and potentially mislead the model. In the following we propose an alternative and demonstrate in the experimental section that this leads to significantly better forecasting performance.

4.3 Adjustable Context-aware Attention

Our goal is to provide the model with the flexibility to choose the optimal context length L for the temporal attention. In the following we refer to optimality with respect to the overarching model's hold-out performance of forecasting target variables, via the query-key similarity.

Rather than learning a single one-size-fits-all context length parameter L, we hypothesize that a successful model would need to switch between different lengths dynamically, depending on the situation. For example, as depicted in Figure 2 for the watershed domain, behavior of the target variable is much more dynamic during storm events than during dry periods. Hence the ideal context length would be much shorter during a storm event than during dry periods. Hence, we will consider multiple context sizes $L = \{l_1, \ldots l_n\}$ when computing the attention score. We will make the selection of the ideal context length part of the prediction problem using the following model.

for all context length l 2 L obtain:

$$\text{query vector } \boldsymbol{q}_{i,l} = \texttt{proj}\left(\boldsymbol{x}_{i_{<l}}\right)$$

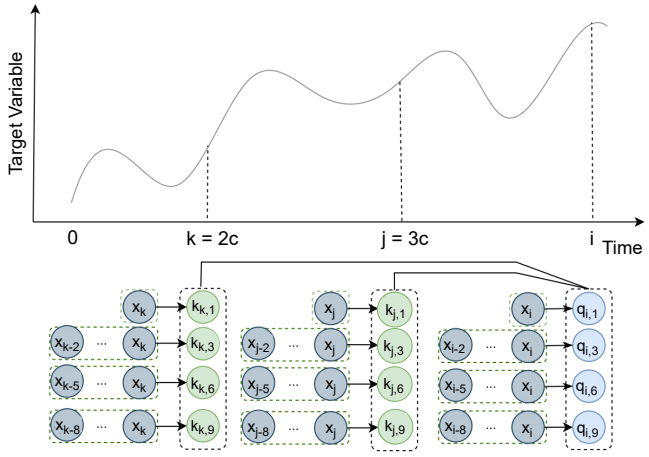


Fig. 4. An overview of our subsampling scheme with $c = 9$. Context sizes are from $\mathcal{L} = \{1, 3, 6, 9\}$. Each key with context length $l \in \mathcal{L}$ subsumes the information of its l preceding inputs. Hence, even though keys from time step $j-1$ to $j-8$ are skipped, the information of x_{j-1} to x_{j-8} is represented in key $k_{j,9}$ via the context-aware attention.

$$\text{key vector } \boldsymbol{k}_{j,l} = \texttt{proj}\left(\boldsymbol{x}_{j<l}\right).$$

Then use the context length that maximizes the attention score:

$$\boldsymbol{a}_{ij} = \max_{l \,2\, L} \texttt{softmax}\left(\boldsymbol{q}_{i,l}^{|} \quad \boldsymbol{k}_{j,l} / \sqrt{p}_{\bar{d}}\right).$$

The intuition is that all wrong filter lengths will miss to detect the similarity, therefore if a filter length triggers a similarity, during training, the back propagation encourages a low similarity across all filter lengths for intervals with different temporal behavior. This adjustable context-aware attention score is used inside the transformer model. Note that, with this approach, the query $\boldsymbol{q}_i$ and the key $\boldsymbol{k}_j$ are represented by multiple context-aware query and key vectors for different context lengths in $\mathcal{L}$. Figure 3 depicts an overview of our proposed attention mechanism.

We claim that this provides the attention mechanism with the flexibility to dynamically determine the optimal query-key similarity and hence leads to better forecasting results. We will provide empirical evidence for this claim in the experimental evaluation.

The downside of this approach is its demand in resources. To calculate the attention weights, our model needs to explore all possible context-aware query and key pairs. For an attention model with Q queries and K keys, computing the attention weights requires $O(|\mathcal{L}|\ Q\ K)$ space and time — not accounting for the cost of projections. This is in contrast to $O(Q\ K)$ for the original point-wise attention. We address this next.

4.4 Efficient Adjustable Context-aware Attention

In order to increase the efficiency our adjustable context-aware attention, we first propose a subsampling scheme for the keys used for attention then explain how our model still incorporates the information from skipped time points.

Let c be the subsampling rate, our transformer would only consider keys with index $j \in \mathcal{J}_c = [0, c, 2c, 3c, \dots]$. For example $c = 2$ would skip every other key, where with $c = 5$, only every 5'th key is considered. However, multiple filter lengths are available at each of these indices, therefore, as an example the information between 0 and c is represented via $\boldsymbol{k}_c$.

Hence the predicted output $\boldsymbol{y}_i$ at time i is obtained as:

$\boldsymbol{y}_i = \mathtt{softmax}(\sum_{j \in \mathcal{J}_c} \boldsymbol{a}_{ij} \boldsymbol{v}_j)$.

In neural networks with point-wise attentions, such a subsampling approach would potentially degrade the performance, as each of the skipped time steps might be potentially crucial for an accurate forecasting result.

Due to the use of the temporal attention in our model, data between skipped keys, for example data from $2c + 1$ to $3c \quad 1$ is represented via $\boldsymbol{k}_{3c}$. This is also depicted in Figure 4. In contrast, if we would not subsample keys, this would lead to lots of redundancies as any input $\boldsymbol{x}_j$ is incorporated into multiple key vectors $\boldsymbol{k}_j, \boldsymbol{k}_{j+1}, \boldsymbol{k}_{j+2} \dots$ due to the temporal attention paradigm.

Context lengths $\mathcal{L}$ and subsampling rate c have opposing effects on the overall complexity, which renders the total memory usage and runtime to $O(\frac{i|\mathcal{L}|}{c} \; Q \; K)$. While $\mathcal{L}$ and c can be independently chosen, in our experimental setup, we use $c = \max \mathcal{L}$. The effect on the overall network is that for the query at time step i and the key at time step j, the network selects the context length l that is sufficient to identify whether the temporal behavior of time step j is helpful to forecast at time step i.

In other words, during periods where a short context is sufficient to identify that the behavior is different, a small context window will be chosen and c keys will be skipped as it is identified as not helpful. Also whenever the similarity is only apparent when inspecting longer contexts, the longer context will be chosen by the network and c keys can be skipped to avoid redundancies. We want to remark that this approach is designed to work with target variables that are smooth over time, with signals that can exhibit rapid or slow changes.

4.5 Overarching Architecture

Our model adapts an encoder-decoder architecture used by Vaswani et al [30].

Encoder. While a stack of ACAT layers could be used, we use an encoder of only a single layer. The encoder is comprised of a multi-head adjustable context-aware self-attention and a feed forward network sub-layer. The encoder is used to encode the information of previous observation (including the target variables) into hidden representations.

Decoder. The decoder is a single layer comprised of a masked multi-head adjustable context-aware self-attention, a multi-head adjustable context-aware cross-attention, and a feed forward network sub-layer. Masked multi-head attention is applied by setting masked dot-products to $-\infty$. This prevents the current time step from attending to future time steps. The decoder generates predictions based on the encoder's hidden representation of previous observations and current known inputs. Alternatively a stack of ACAT decoder layers could be used, here we demonstrate the efficacy of even a single layer.

5 Experiments

5.1 Datasets

We empirically perform experiments on three datasets, including two univariate publicly available datasets and one multivariate dataset that we provide in our github repository.

Electricity:[2] The univariate UCI Electricity Load Diagrams dataset, containing the electricity consumption of 370 customers aggregated on an hourly level.
Traffic:[3] The univariate UCI PEM-SF Traffic Dataset, containing occupancy rate ($y_t \in [0, 1]$) of 440 SF Bay Area freeways aggregated on an hourly level.
Watershed:[4] This multivariate dataset contains hydrological streamflow responses of ten watershed sites, aggregated on a 15 minutes level.

Regarding our choices of datasets, we select the traffic and electricity datasets that have been extensively used by a significant amount of research papers for modeling and evaluation [25, 20, 18, 11, 33]. We are also working with collaborators who are interested in modeling real-world stream chemistry for the watershed dataset. We use 160,000 samples for each dataset, each sample including historical observations of one week (168 hours) for the traffic and electricity datasets and 42 (168 quarter) hours for the watershed dataset. We generate predictions for 24 and 48 future horizons on all datasets. After zero-mean normalization, we partition each dataset into three parts, 80% training set for the learning procedure, 10% validation set for hyperparameter optimization, and 10% hold-out test set for performance evaluation.

5.2 Evaluation Metrics

Models are evaluated by two standard metrics, including root mean squared error (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2}$ and mean absolute error (MAE): $\frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$. MAE is a linear score that equally weighs the errors, where RMSE is a quadratic score that assigns higher weights to larger errors.

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/00321/LD2011-2014
[3] https://archive.ics.uci.edu/ml/machine-learning-databases/00204
[4] https://github.com/a1992/Context-Aware-Transformer/data/watershed

Table 1. Results summary in RMSE and MAE of all methods on three datasets. Lower RMSE and MAE indicate a more accurate forecast. Best results are highlighted in boldface.

Dataset	Horizon	Metric	ARIMA		LSTM		Transformer		Trans-multi		CNN-trans		ACAT (Ours)	
Traffic	24	RMSE	0.81	0.00	0.50	0.02	0.48	0.00	0.59	0.09	0.47	0.00	**0.38**	0.00
		MAE	0.56	0.00	0.28	0.02	0.25	0.00	0.35	0.09	0.24	0.00	**0.16**	0.00
	48	RMSE	0.79	0.00	0.49	0.00	0.46	0.00	0.68	0.09	0.46	0.00	**0.35**	0.00
		MAE	0.56	0.00	0.26	0.00	0.24	0.00	0.45	0.09	0.23	0.00	**0.16**	0.00
Electricity	24	RMSE	3.98	0.00	1.29	0.03	1.29	0.07	1.48	0.08	1.27	0.08	**0.64**	0.02
		MAE	0.41	0.00	0.13	0.01	0.15	0.01	0.16	0.00	0.14	0.00	**0.08**	0.00
	48	RMSE	4.06	0.00	1.40	0.07	1.47	0.02	1.56	0.14	1.26	0.04	**0.84**	0.03
		MAE	0.41	0.00	0.14	0.00	0.16	0.00	0.17	0.00	0.14	0.00	**0.09**	0.00
Watershed	24	RMSE	-		0.35	0.05	0.33	0.01	0.35	0.01	0.34	0.00	**0.28**	0.01
		MAE			0.20	0.03	0.19	0.01	0.21	0.01	0.20	0.00	**0.16**	0.01
	48	RMSE	-		0.42	0.04	0.36	0.01	0.35	0.01	0.35	0.01	**0.31**	0.01
		MAE			0.25	0.01	0.21	0.03	0.20	0.01	0.20	0.01	**0.16**	0.01

5.3 Baselines

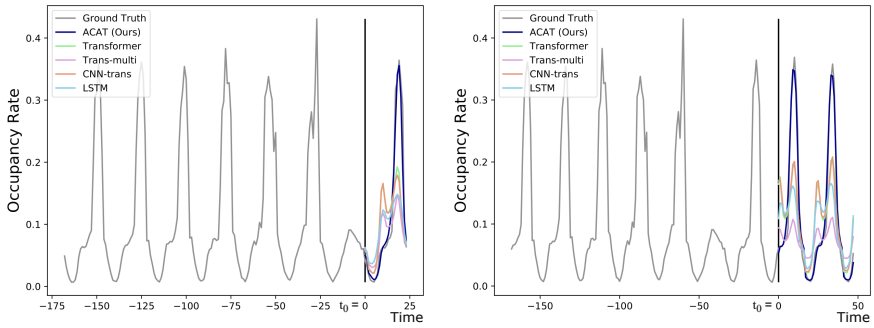
We compare a single layer architecture of our proposed ACAT model to the following methods. ARIMA is only applicable to univariate datatsets (traffic and electricity).

1. **ARIMA [3]**: Auto-regressive integrated moving average.
2. **LSTM [13]**: A single layer encoder-decoder Long Short Term Memory.
3. **Transformer [30]**: A single layer transformer equivalent to our approach with the basic multi-head attention.
4. **Trans-multi [30]**: A three encoder layer and one decoder layer transformer with multi-head basic attention.
5. **CNN-trans [18]**: A single layer transformer with convolutional multi-head attention.

5.4 Model Training and Hyperprarameters

Training Procedure. All neural network methods are trained and evaluated multiple times. We use grid search for hyperparameter tuning. The model size is chosen from $f16, 32g$ for all neural network methods. The number of heads is set to 8 for all transformer-based models. The kernel size for convolutional processing layer of CNN-trans is chosen from $f1, 3, 6, 9g$. The mini batch-size is set to 256 for all datasets. We use the Adam optimizer [16] with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$, we update the learning rate and use $\texttt{warmup\_steps} = 4000$ according to the basic transformer [30]:

$$\texttt{lrate} = d_{\texttt{model}}^{-0.5} . \min(\texttt{step\_num}^{-0.5}, \texttt{step\_num} . \texttt{warmup\_steps}^{-1.5})$$



a) 24 horizons b) 48 horizons

Fig. 5. Forecasted predictions of one sample of hold-out test set on the traffic dataset of neural network models. Predictions are generated for a) 24 and b) 48 horizons given the previous 168 input samples. The predictions generated by our ACAT model are exhibiting a higher accuracy than other baselines.

The total number of epochs is 50 with early stopping set to five iterations.

Loss Function. For all methods, we choose the mean squared error (MSE) loss function to calculate the loss of generated predictions compared to the target sequence in the training procedure. The loss is back-propagated from the decoder's outputs to the entire model.

Hardware and Computational Cost. All models were trained on a single NVIDIA Titan XP GPU with 12 GB of memory. For our proposed ACAT model with $L = \{1, 3, 6, 9\}$, it takes 0.3 seconds to finish one training step and each epoch takes 150 seconds.

5.5 Results and Discussion

Results are reported as mean and standard error of RMSE and MAE scores for the total number of three experimental runs for neural networks and one experimental run for the ARIMA model.

Table 1 summarizes the evaluation results of all methods on three datasets when generating predictions for 24 and 48 forecasting horizons. Across all datasets we observe that ACAT outperforms other methods. The significant improvements of our ACAT model over other methods stem from our proposed adjustable context-aware attention. The difference in performance to CNN-trans demonstrates the gain when providing the model with the flexibility to choose the right context length adaptively. Regarding baselines, the performance of the Trans-multi model indicates that even a multi-layer transformer is ineffective in recovering from the point-wise attention. The performance of the CNN-trans

Table 2. Comparison of different subsampling rates. As a result of increasing the subsampling rate (maximum context size $\max(\mathcal{L})$) more keys are skipped. Lower RMSE and MAE indicate a more accurate forecast. Best results are highlighted in boldface. The last row indicates how many times this configuration obtained the best result. The performance of our ACAT model is resilient towards the subsampling rate.

Dataset	Horizon	Metric	$\max(\mathcal{L})=9$		$\max(\mathcal{L})=12$		$\max(\mathcal{L})=15$		$\max(\mathcal{L})=18$		$\max(\mathcal{L})=21$		$\max(\mathcal{L})=24$	
Traffic	24	RMSE	0.38	0.00	0.38	0.00	**0.37**	0.00	**0.37**	0.00	**0.37**	0.00	0.38	0.00
		MAE	0.16	0.00	0.16	0.00	0.16	0.00	**0.15**	0.00	0.16	0.00	**0.15**	0.00
	48	RMSE	**0.35**	0.00	0.36	0.00	0.36	0.00	0.37	0.00	0.37	0.00	0.37	0.00
		MAE	**0.16**	0.00	**0.16**	0.00	**0.16**	0.00	0.17	0.00	0.17	0.00	**0.16**	0.00
Electricity	24	RMSE	0.64	0.02	0.66	0.02	**0.63**	0.02	0.67	0.03	0.73	0.03	0.72	0.05
		MAE	**0.08**	0.00	**0.08**	0.00	**0.08**	0.00	**0.08**	0.00	**0.08**	0.00	0.09	0.00
	48	RMSE	0.84	0.03	0.84	0.01	0.83	0.03	**0.80**	0.09	0.83	0.06	0.83	0.01
		MAE	**0.09**	0.00	**0.09**	0.00	**0.09**	0.00	**0.09**	0.01	**0.09**	0.00	**0.09**	0.00
Watershed	24	RMSE	0.28	0.01	**0.25**	0.00	0.28	0.01	0.27	0.00	0.27	0.00	0.28	0.02
		MAE	0.16	0.01	**0.13**	0.00	0.15	0.01	0.14	0.00	0.16	0.00	0.15	0.00
	48	RMSE	0.31	0.01	0.30	0.01	0.31	0.02	0.30	0.01	**0.29**	0.00	0.30	0.01
		MAE	**0.16**	0.00	0.17	0.01	0.17	0.01	**0.16**	0.01	**0.16**	0.00	**0.16**	0.01
Total Wins			5		5		5		6		5		4	

model is relatively similar to the basic transformer. This indicates that integrating a temporal context with a fixed length cannot enhance the forecasting quality considerably if the model is not able to adjust the context length.

Figure 5 displays forecasted predictions of all neural network methods including our ACAT on one example from traffic dataset over 24 and 48 forecasting horizon. The predictions generated by ACAT are aligning closely with the ground truth, while the generated predictions of other neural network baselines exhibit larger errors and fail to resemble the temporal behaviour.

Ablation Study: The Impact of Subsampling Rate. To demonstrate that the performance of our ACAT model is resilient towards how many keys are skipped during subsampling, we conduct an ablation study by increasing the subsampling rate. As a result of choosing the subsampling rate as the maximum context size $\max(\mathcal{L})$, we skip more keys by increasing the value of the maximum context size. Table 2 demonstrates the results of this ablation study, it is observed that the performance of our model is resilient towards the subsampling rate. In contrast, increasing the subsampling rate allows us to include a longer history into the forecasting while adjusting the temporal context length, which leads to a better performance in some cases.

6 Conclusion

In this paper, we study the multi-horizon time series forecasting problem and introduce ACAT, the Adjustable Context-aware Transformer, which automatically selects the ideal context size to obtain the best forecasting results. We demonstrate the effectiveness of our approach on three real-world datasets in comparison with the classical forecasting method ARIMA, the RNN-based encoder-decoder LSTM, transformers based with basic and CNN attention, as well as multi-layer transformers. Our extensive analyses show that our ACAT model obtains performance improvements over state-of-the-art temporal attention approaches, including those based on the convolutional attention models. This indicates that incorporating the ideal context length in the query-key similarity of the attention mechanism can improve the forecasting quality.

In the introduction we hypothesized that the structure of the basic attention mechanism presents an impediment for the original transformer architecture when applied to temporal analyses, even when including multiple layers. Experimentally we verify this claim, and demonstrate that our ACAT model addresses its shortcoming even with a single layer.

Lastly, our ACAT model can benefit any application domain where accurate forecasts are important. For example in the watershed domain, natural scientists base their analysis on forecasting models. Incorrect forecasts can lead to false models of natural processes. Another example are forecasts on traffic or electricity consumption, which, if inaccurate, have severe negative effects on our society. Our work demonstrates significant improvements in all these three domains.

References

[1] Ahmed M. Alaa and Mihaela van der Schaar. "Attentive State-Space Modeling of Disease Progression". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/1d0932d7f57ce74d9d9931a2c6db8a06-Paper.pdf.

[2] Sercan Ö. Arik and Tomas Pfister. "TabNet: Attentive Interpretable Tabular Learning". In: *Proceedings of the AAAI Conference on Arti cial Intelligence* 35.8 (May 2021), pp. 6679–6687. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16826.

[3] G. E. P. Box and G. M. Jenkins. "Some Recent Advances in Forecasting and Control". In: *Journal of the Royal Statistical Society Series C* 17.2 (June 1968), pp. 91–109. DOI: 10.2307/2985674. URL: https://ideas.repec.org/a/bla/jorssc/v17y1968i2p91-109.html.

[4] Carlos Capistrán, Christian Constandse, and Manuel Ramos-Francia. "Multi-horizon inflation forecasts using disaggregated data". In: *Economic Modelling* 27.3 (May 2010), pp. 666–677. URL: https://ideas.repec.org/a/eee/ecmode/v27y2010i3p666-677.html.

[5] Yubo Chen et al. "Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 167–176. DOI: 10.3115/v1/P15-1017. URL: https://aclanthology.org/P15-1017.

[6] Edward Choi et al. "RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 3512–3520. ISBN: 9781510838819.

[7] Pascal Courty and Hao Li. "Timing of Seasonal Sales". In: *The Journal of Business* 72.4 (Oct. 1999), pp. 545–572. DOI: 10.1086/209627. URL: https://ideas.repec.org/a/ucp/jnlbus/v72y1999i4p545-72.html.

[8] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. OUP Catalogue 9780198523543. Oxford University Press, 2001. ISBN: ARRAY(0x4eed11c8). URL: https://ideas.repec.org/b/oxp/obooks/9780198523543.html.

[9] Chenyou Fan et al. "Multi-Horizon Time Series Forecasting with Temporal Attention Learning". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2527–2535. ISBN: 9781450362016. DOI: 10.1145/3292500.3330662. URL: https://doi.org/10.1145/3292500.3330662.

[10] Alex Graves. "Generating Sequences With Recurrent Neural Networks." In: *CoRR* abs/1308.0850 (2013). URL: http://dblp.uni-trier.de/db/journals/corr/corr1308.html#Graves13.

[11] haixu wu haixu et al. "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 22419–22430. URL: https://proceedings.neurips.cc/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf.

[12] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[13] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[14] Cheng-Zhi Anna Huang et al. "Music Transformer: Generating Music with Long-Term Structure". In: *ICLR*. 2019.

[15] Urvashi Khandelwal et al. "Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context". In: *Proceedings of the 56th Annual Meet-*

ing of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 284–294. DOI: 10.18653/v1/P18-1027. URL: https://aclanthology.org/P18-1027.

[16] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[17] Yann LeCun and Yoshua Bengio. "Convolutional Networks for Images, Speech, and Time Series". In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, pp. 255–258. ISBN: 0262511029.

[18] Shiyang Li et al. "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[19] Bryan Lim. "Forecasting Treatment Responses Over Time Using Recurrent Marginal Structural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/56e6a93212e4482d99c84a639d254b67-Paper.pdf.

[20] Bryan Lim et al. "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL: https://www.sciencedirect.com/science/article/pii/S0169207021000637.

[21] Jiawei Ma et al. *CDSA: Cross-Dimensional Self-Attention for Multivariate, Geo-tagged Time Series Imputation*. 2019. arXiv: 1905.09904 [cs.LG].

[22] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: 100,000 time series and 61 forecasting methods". In: *International Journal of Forecasting* 36.1 (2020), pp. 54–74. URL: https://EconPapers.repec.org/RePEc:eee:intfor:v:36:y:2020:i:1:p:54-74.

[23] Spiros Papadimitriou and Philip Yu. "Optimal multi-scale patterns in time series streams". In: Jan. 2006, pp. 647–658. DOI: 10.1145/1142473.1142545.

[24] Syama Sundar Rangapuram et al. "Deep State Space Models for Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf.

[25] David Salinas et al. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks". In: *International Journal of Forecasting* 36.3 (2020), pp. 1181–1191. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2019.07.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301888.

[26] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. "Temporal pattern attention for multivariate time series forecasting". In: *Machine Learning* 108 (Sept. 2019). DOI: 10.1007/s10994-019-05815-0.

[27] Huan Song et al. "Attend and Diagnose: Clinical Time Series Analysis Using Attention Models". In: AAAI Press, 2018. ISBN: 978-1-57735-800-8.

[28] Pengfei Tang et al. "Channel Attention-Based Temporal Convolutional Network for Satellite Image Time Series Classification". In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. DOI: 10.1109/LGRS.2021.3095505.

[29] Sean Taylor and Benjamin Letham. *Forecasting at scale*. Sept. 2017. DOI: 10.7287/peerj.preprints.3190v2.

[30] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[31] Fei Wang et al. "Residual Attention Network for Image Classification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6450–6458. DOI: 10.1109/CVPR.2017.683.

[32] Ruofeng Wen et al. "A multi-horizon quantile recurrent forecaster". In: *NeurIPS 2017*. 2017. URL: https://www.amazon.science/publications/a-multi-horizon-quantile-recurrent-forecaster.

[33] Haoyi Zhou et al. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *AAAI*. 2021.

[34] Yunyue Zhu and Dennis Shasha. "StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time". In: *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB '02. Hong Kong, China: VLDB Endowment, 2002, pp. 358–369.