Exploring the Role of Token in Transformer-based Time Series Forecasting

Jianqi Zhang[1], Jingyao Wang[1], Chuxiong Sun[1], Xingchen Shen[1], Fanjiang Xu[1], Changwen Zheng[1], Wenwen Qiang[1]

[1]University of Chinese Academy of Sciences, Beijing, China.
jluzhangjianqi@163.com, wangjingyao2023@iscas.ac.cn, xingchen@iscas.ac.cn, fanjiang@iscas.ac.cn, changwen@iscas.ac.cn, qiangwenwen@iscas.ac.cn

Abstract

Transformer-based methods are a mainstream approach for solving time series forecasting (TSF). These methods use temporal or variable tokens from observable data to make predictions. However, most focus on optimizing the model structure, with few studies paying attention to the role of tokens for predictions. The role is crucial since a model that distinguishes useful tokens from useless ones will predict more effectively. In this paper, we explore this issue. Through theoretical analyses, we find that the gradients mainly depend on tokens that contribute to the predicted series, called positive tokens. Based on this finding, we explore what helps models select these positive tokens. Through a series of experiments, we obtain three observations: i) positional encoding (PE) helps the model identify positive tokens; ii) as the network depth increases, the PE information gradually weakens, affecting the model's ability to identify positive tokens in deeper layers; iii) both enhancing PE in the deeper layers and using semantic-based PE can improve the model's ability to identify positive tokens, thus boosting performance. Inspired by these findings, we design temporal positional encoding (T-PE) for temporal tokens and variable positional encoding (V-PE) for variable tokens. To utilize T-PE and V-PE, we propose T2B-PE, a Transformer-based dual-branch framework. Extensive experiments demonstrate that T2B-PE has superior robustness and effectiveness.

Introduction

Time series forecasting (TSF) is essential in numerous practical applications, e.g., weather forecasting (Abhishek et al. 2012; Karevan and Suykens 2020; Agrawal et al. 2012), energy planning (Debnath and Mourshed 2018; Boussif et al. 2024; Novo et al. 2022) and traffic flow forecasting (Fang et al. 2023; Li et al. 2022; Wang et al. 2022). Recently, Transformer-based methods (Wu et al. 2020; Lim et al. 2021; Liu et al. 2023) have become one of the mainstream approaches and achieved good results. They take two types of tokens as input: (i) temporal tokens, which contain all variables of the same timestamp; (ii) variable tokens, which contain all input time points for a specific variable. Despite their empirical effects being widely proven, few works explore the role of tokens for prediction. This role is crucial because if the model can identify it, it will be easier for the model to achieve better prediction results (Figure 3). In this paper, we investigate this issue from the perspective of model optimization.

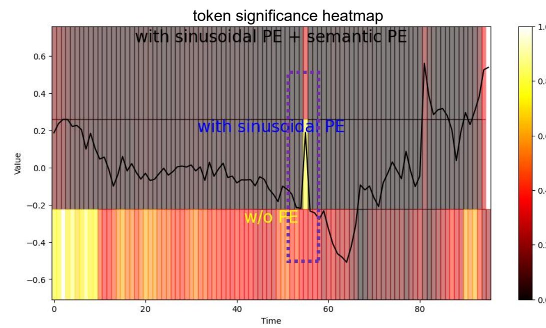


Figure 1: The visualizations of input token significance in the first attention layer of the Transformer under three conditions: i) with sinusoidal PE and semantic PE (top, MSE: 0.27); ii) with sinusoidal PE only (middle, MSE: 0.32); iii) without PE (bottom, MSE: 0.85). The black line represents the input series. The peak in the purple dashed box corresponds to a noise token.

Firstly, through a series of theoretical derivations in the "Theoretical Analysis" section, we find that during the training process, the gradients of the Transformer-based time series forecasting model primarily depend on positive tokens. We define input tokens closely related to the predicted series as "positive tokens" and, conversely, as "negative tokens." We visualize the model's identification of positive tokens by plotting the token significance as a heatmap (Figure 1). The significance is calculated Eq.2. In Figure 1, tokens with a higher significance are recognized as positive tokens by the model. Ideally, in Figure 1, there are two types of temporal tokens that are more likely to be positive: i) the last few tokens in the input series and ii) tokens at peaks or valleys. The former is because these tokens are closely related to the predicted series; the latter is because peaks and troughs typically contain strong temporal information (Dong et al. 2023). The model updates in the gradient direction, which depends on positive tokens. Thus, the model primarily learns from positive tokens. However, the model's identified positive tokens aren't always accurate. To understand what influences the model's selection of positive tokens, we conducted some experimental explorations.

We conduct four sets of experiments to explore the above issue. Considering that positional encoding (PE) can help the model identify the position of tokens, including peaks or valleys, we conduct experiments to evaluate the effect of PE. Specifically, in **the first experiment**, we train the Transformer with and without sinusoidal PE and then visualize token significance (Figure 1). The results show that the model without PE will randomly assign high significance to many tokens; when PE is used, important tokens, e.g., the last few tokens and tokens in peaks, are well identified. It indicates that PE helps the model identify positive tokens. Ideally, the model should better perceive positional information from PE in deeper layers to more accurately identify positive tokens. However, in **the second experiment**, we find that the PE information weakens in deeper layers, affecting the models' ability to identify positive tokens (Figure 3). Specifically, we evaluate the Mutual Information (MI) (Steuer et al. 2002) between the PE and the input embedding of each attention layer of multiple models and find the MI decreases as network depth increases. This inspired us to enhance PE in deeper layers, named enhanced PE. In **the third experiment**, we add the enhanced PE into the multiple models and find it alleviates the problem mentioned above and improves the model's performance (Table 1). The PEs discussed above are based on geometric positions, referred to as geometric PEs. However, geometric PEs cannot distinguish between tokens that are close in position but differ in value, which makes the model with sinusoidal PE assign a high significance to the noise token in the purple dashed box of Figure 1. To avoid this, we design a PE based on the tokens' semantics called semantic PE. To evaluate its effect, in **the fourth experiment**, we add the semantic PE into the multiple models and find it helps identify positive tokens, i.e., decrease the significance of noise token in Figure 1) and improve the models' performance (Table 1), which indicates the effectiveness of the semantic PE. Details of these four experiments are in the "Empirical Analysis" section.

Based on the above observations, we propose two new types of PE: temporal positional encoding (T-PE) for the temporal tokens and variable positional encoding (V-PE) for the variable tokens. Both T-PE and V-PE consist of a geometric PE and a semantic PE. The semantic PEs in T-PE and V-PE are the same. The geometric PE in T-PE is based on the enhanced sinusoidal PE, while the geometric PE in V-PE is based on convolution. This difference arises because the two types of tokens have different requirements for positional information, as discussed in the "Motivation of T-PE and V-PE" section. To leverage both T-PE and V-PE, we develop a novel Transformer-based dual-branch framework called T2B-PE. T2B-PE first separately handles the correlations between temporal tokens and the correlations between variable tokens. Then, it combines the dual-branch features using a novel gated unit designed for TSF to predict the results. Each branch utilizes the encoder of the vanilla Transformer with T-PE or V-PE. In the experimental section, we validate the effectiveness of T2B-PE using six real-world datasets and further confirm the effectiveness of T-PE and V-PE through extensive visualization experiments. Our contributions are as

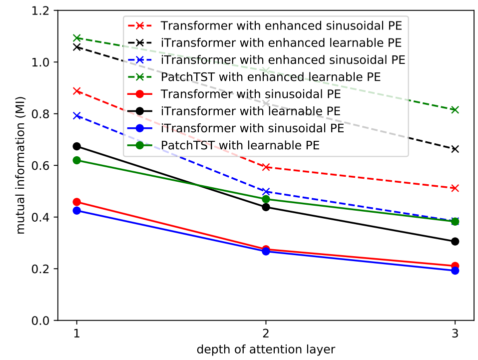


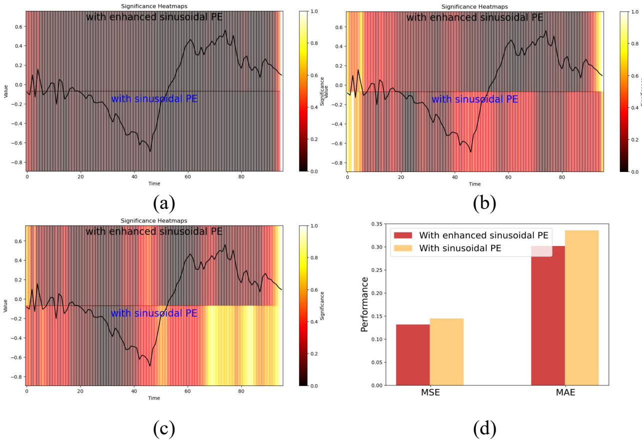
Figure 2: Illustration of the change in the Mutual Information (MI) between the PE and the input embedding of each attention layer with the depth of the attention layer. The solid line represents the results of the method with the normal PE, while the dashed line represents the results with our enhanced PE. For details on MI, see Appendix D.4.

follows:

- We are the first to explore the role of tokens for prediction in Transformer-based TSF and theoretically find that the gradient of the Transformer-based TSF model mainly depends on positive tokens during training.
- We conduct extensive empirical analyses and conclude: i) PE helps the model identify positive tokens; ii) as the network depth increases, the PE information gradually weakens, affecting performance; iii) geometric PE and semantic PE can both enhance the model's ability to identify positive tokens and improve performance.
- We propose two novel PEs for TSF: T-PE for the temporal tokens and V-PE for the variable tokens. To leverage both T-PE and V-PE, we propose a simple but powerful Transformer-based dual-branch framework (T2B-PE).
- We conduct extensive experiments on six benchmark datasets, demonstrating the superior robustness and effectiveness of the proposed T2B-PE and two new PEs.

Related Work

Transformer-based TSF methods: Transformer-based methods are one of the mainstream approaches for TSF. Currently, many improvements have been made to these methods. Some methods (Wu et al. 2021a; Zhou et al. 2022; Liang et al. 2023) combine Fourier transform with Transformer to leverage the periodicity of time series. GCformer (Zhao et al. 2023) integrates convolution with Transformer. Others (Du, Su, and Wei 2023; Tang and Zhang 2023) enhance the attention mechanism to improve TSF performance. PatchTST (Nie et al. 2023) embeds multiple adjacent time points into one temporal token to address insufficient temporal information of a single temporal token. iTransformer shows that using only variable tokens can yield good results. Crossformer (Zhang and Yan 2023) and DSformer (Yu et al. 2023) utilize both temporal and variable tokens for TSF. Among



(a) (b) (c) (d)

Figure 3: The visualizations of input token significance in the three attention layers of the Transformer under two conditions: i) with enhanced sinusoidal PE; ii) with sinusoidal PE. (a) The token significance of the first layer. (b) The token significance of the second layer. (c) The token significance of the third layer. (d) Prediction performance. (a), (b) and (c) show that as the network deepens, the model's judgment of positive tokens becomes more confused.

these methods, only iTransformer does not use PE, but experiments show that adding PE to its variable tokens improves its performance (Table 1). Other methods include sinusoidal PE or learnable PE from NLP. These methods mainly focus on optimizing model architecture, with little research on the role of to- kens for predictions.

PE in Transformer-based methods: There are many articles on PE in Transformer-based methods, but most of them focus on other fields such as natural language processing (Su et al. 2024; Shaw, Uszkoreit, and Vaswani 2018; Ke, He, and Liu 2020), computer vision (Chu et al. 2021; Wu et al. 2021b; Jiang et al. 2022), and time series classification (Foumani et al. 2024). In TSF, only sinusoidal PE or scientific PE from NLP is typically used, and PE research is still insufficient. Currently, the only available study is Seformer (Zeng et al. 2023b). However, Seformer simply adapts the PE from other fields for TSF without thoroughly analyzing the issues of the commonly used PE in TSF.

Unlike previous methods, this paper explores the role of tokens for predictions from an optimization perspective, revealing that the model's gradients are primarily associated with positive tokens. To understand what influences the model's ability to recognize positive tokens, we conduct experimental explorations and find that PE helps the model recognize positive tokens. We also find that enhanced PE and semantic PE are particularly effective in this role. Based on these findings, we design two new types of PE: T-PE and V-PE.

Theoretical Analysis

According to (Dai et al. 2021), every neuron in the Transformer-based model has the incentive to be positively correlated with the predicted time series. Meanwhile, the positive tokens correlate strongly with the predicted time series. To explore the relationship between neurons and positive tokens, we get:

Theorem 1. *Given a Transformer-based model f_θ for TSF and the training data $\mathcal{D}_{tr} = (X, Y)$, it is trained via SGD to minimize the loss $\ell(Y, f(X))$ with step size σ for T iterations. Here, X denotes the input tokens and Y denotes the future timestamps that aim to predict in TSF. For every $\sigma \frac{1}{\Theta(d^C)}$ and every $y \in Y$ where C is unspecified large constant and d is the number of features, w.h.p., for every iteration $t \geq T^{ini} = \Theta(\frac{m}{\sigma\sqrt{d}})$, there exist $\Theta(m)$ neurons in which the weight w_i^t for each neuron satisfies:*

$$P_{x|y\sim\mathcal{D}_{tr}}\left[\langle w_i^t, x\rangle \geq 0\right] = 1 - o(d^{-\frac{1}{3}})$$
$$P_{x|-y\sim\mathcal{D}_{tr}}\left[\langle w_i^t, x\rangle \geq 0\right] = o(1) \tag{1}$$

where o is the standard little-o notation, y and $-y$ corresponding to positive and negative tokens respectively.

Through this theorem, we can see that the model weight in each iteration is strongly related to the positive token, i.e., $1 - o(d^{-\frac{1}{3}}) \approx 1$, and has weak relation with the negative token, i.e., $o(1) \approx 0$. After enough SGD iterations, all neurons in the model will (as expected) rely on and benefit from positive tokens for predictions while accumulating negative correlations with observations from negative tokens. This indicates that the gradient of neurons in the Transformer-based model for TSF mainly depends on the positive token of the neuron, i.e., it gets a positive gradient projection but has almost nothing to do with the negative token. The details proof is provided in Appendix E.

Empirical Analysis

The positive tokens identified by the model are not always accurate. To understand what influences the model's selection of positive tokens, this section conducts four experiments.

In the first experiment, we explore the impact of PE on the model's ability to identify positive tokens. Specifically, we train the Transformer on the Weather dataset (Liu et al. 2023) under two conditions: without PE, with the sinusoidal PE. We record the token significance in the first layer of the trained model on the test dataset and plot it as a heatmap (Figure 1). Token significance is calculated as follows:

$$\mathbf{Sign}(x_i) = \sum_{j=1}^{T} \mathbf{Att_{j,i}}, \quad i = 1, 2, ...T, \tag{2}$$

where x_i is the i-th token, T is the number of tokens, $\mathbf{Att} = \mathbf{softmax}(QK^\top/\sqrt{D}) \in \mathbb{R}^{T\times T}$ is the attention map, D is the dimension of tokens. Due to the softmax, the sum of each row of $\mathbf{Att}$ is 1, while the sum of each column is the token significance $\mathbf{Sign}$. Tokens with large token significance have a strong correlation with the predicted series and are positive tokens, while those with low token significance are negative tokens. From Figure 1, We observe: i) without PE, the model assigns a very high token significance to most tokens.; ii) with sinusoidal PE, the significance of tokens at the last few positions and peaks is high. As discussed in the Introduction, in Figure 1, tokens at the last few positions and peaks contribute more significantly to the prediction. Thus, this indicates that PE helps the model identify positive tokens.

Table 1: Experimental results on the Weather dataset and ECL dataset with different PEs. For more details, please refer to Appendix F.

MSE/MAE	Dataset	Original	+enhanced PE	+semantic PE
Transformer+sinusoidal PE	Weather	0.379/0.432	**0.339/0.393**	0.347/0.400
	ECL	0.333/0.414	**0.289/0.381**	0.313/0.391
PatchTST+learnable PE	Weather	0.259/0.281	**0.250/0.274**	0.255/0.276
	ECL	0.216/0.304	**0.211/0.298**	0.213/0.300
iTransformer+sinusoidal PE	Weather	0.255/0.278	0.254/0.277	**0.248/0.276**
	ECL	0.174/0.267	0.171/0.266	**0.168/0.263**
iTransformer+learnable PE	Weather	0.253/0.278	0.250/0.276	**0.248/0.275**
	ECL	0.176/0.268	0.173/0.267	**0.170/0.265**
iTransformer(w/o PE)	Weather	0.258/0.279	-	-
	ECL	0.178/0.270	-	-

In the second experiment, we explore how the amount of positional information changes as the network deepens. Specifically, we use the Weather dataset and ECL dataset in (Liu et al. 2023) to train three methods: vanilla Transformer (Vaswani et al. 2017), PatchTST (Nie et al. 2023) and iTransformer (Liu et al. 2023) with the sinusoidal PE or the learnable PE. Vanilla Transformer and PatchTST only deal with the temporal tokens, and iTransformer only deals with the variable tokens. In this experiment, the lookback length is 96 and the prediction lengths $\in \{96, 192, 336, 720\}$. The number of encoder layers is set to 3. We compute the MI between the PE and the input embedding of each attention layer. The results in Figure 2 suggest that as the depth of the network increases, there is a decrease in the MI, indicating a decrease in positional information. Meanwhile, we observe that as the network depth increases, the model's identification of positive tokens becomes more chaotic (Figure 3), indicating that the reduction in positional information also affects the model's ability to identify positive tokens.

In the third and fourth experiments, we explore whether the enhanced or semantic PE benefits model performance. Specifically, we use the same experiment settings in the second experiment. Next, after adding the enhanced PE or the semantic PE, we train the models. Then, we record the average mean squared error (MSE), the average mean absolute error (MAE), and the MI (only for the enhanced PE). The results are shown in Table 1 and Figure 2. We observe that i) the enhanced PE mitigates the phenomenon of the decay of the positional information and ii) both the enhanced PE and the semantic PE benefit model performance. Meanwhile, we also recorded the changes in token significance after adding these two PEs. The results show that i) the semantic PE effectively reduces the significance of anomalous tokens (Figure 1), and ii) the enhanced sinusoidal PE allows the model to easily identify the last few tokens as positive in deeper network layers while preventing the model from assigning high significance to most tokens, something that sinusoidal PE cannot achieve. (Figure 3). These results validate the effectiveness of the enhanced PE and the semantic PE.

In this section, we conduct experiments and conclude that i) PE helps the model identify positive tokens and ii) compared to the commonly used PEs in the TSF, enhanced PE and semantic PE are more effective in this aspect. Those findings inspire us to design more effective PEs.

Motivation of T-PE and V-PE

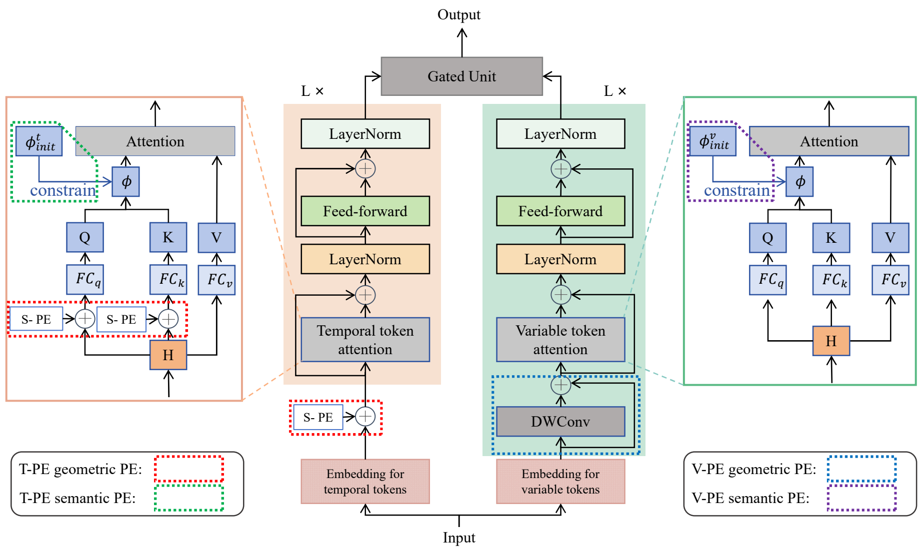
In this section, based on the preceding theoretical and experimental results, we introduce the components of T-PE and V-PE and analyze the motivation of each component.

T-PE consists of a geometric PE and a semantic PE. For the **geometric PE** of T-PE, the intuitive idea is to use the enhanced PE mentioned in the Introduction. Its effectiveness has been proven in Table 1. In general, there are two types of geometric PE in TSF, e.g., sinusoidal and learnable PE. Through ablation experiments in Appendix H, we find that the former performs better. Thus, we use the enhanced sinusoidal PE as the geometric PE of T-PE. For the **semantic PE** of T-PE, we design it based on the semantic similarity of initial tokens, just like geometric PE is based on initial geometric positions."initial" means not influenced by the attention mechanism. Both PEs provide the model with prior knowledge about token proximity. The semantic similarity of initial tokens makes it easier to distinguish between tokens that are close in position but differ significantly in value, which the geometric PE struggles to do.

V-PE also consists of a geometric PE and a semantic PE. To design a good **geometric PE** of V-PE, it is essential to recognize the significant differences between variable tokens and temporal tokens: i) A variable token contains the temporal information of the entire input series, whereas a temporal token only contains that of a time point; ii) Swapping any two variable tokens does not affect the input's semantics (called permutation invariance), but it has an impact on temporal tokens. Note that the permutation invariance of variable tokens does not imply that their positions are unimportant. From our experiments (Table 1), it can be seen that simply adding PE to the variable tokens of iTransformer can improve its performance. The commonly used PEs in TSF (sinusoidal PE and learnable PE) have two main issues: first, they do not utilize the rich temporal information in variable tokens; second, they lack permutation invariance. According to the results shown in Table 1, the enhanced PE does not significantly improve iTransformer based on the variable tokens. This prompts us to develop a new geometric PE for variable tokens that considers both semantics and positions and possesses permutation invariance. Convolution, fortunately, has these two properties (Islam, Jia, and Bruce 2020; Chu et al. 2021). Therefore, we generate the geometric PE of V-PE through convolution. For the **semantic PE** of V-PE, it is evident that the corresponding design of T-PE is also suitable for variable tokens.

Method

In this section, we introduce the details of the proposed T2B-PE. This method features a dual-branch structure, with the left branch processing temporal tokens with T-PE and the right branch processing variable tokens with V-PE. Specifically, we first introduce how to insert T-PE into temporal tokens. Next, we explain how to insert V-PE into variable tokens. Finally, we present the overall process of T2B-PE, which includes T-PE and V-PE. Due to space limitations in the main text, the details of the method are presented in Appendix A.



(a) (b) (c)

Figure 4: Overview of T2B-PE. (a) The details of temporal token attention, where S-PE means the sinusoidal PE, ϕ^t_{init} is the initial token similarity matrix, ϕ is the attention map. (b) The structure of T2B-PE, where DWConv is the convolution used to generate the PE. (c) The details of variable token attention. T2B-PE's left branch is the temporal token branch, and its right branch is the variable token branch. The position of T-PE is marked by the red and green dashed boxes. The position of V-PE is marked by the blue and purple dashed boxes.

As discussed earlier, T-PE consists of a geometric PE and a semantic PE. The **geometric PE** is the enhanced sinusoidal PE. Specifically, the enhanced sinusoidal PE means adding the sinusoidal PE not only at the input layer but also at the input of each attention layer. The locations where this PE is added are masked by the red dashed boxes in Figures 4(a) and 4(b). The **semantic PE**, on the other hand, is based on the initial token similarity ϕ^t_{init}, which has the same shape as the attention map ϕ. Both ϕ^t_{init} and ϕ represent the relational similarity between tokens. Thus, we constrain the L2 distance between the initial token similarity matrix and the attention map to insert the semantic PE. The locations where the semantic PE is inserted are masked by the green dashed boxes in Figures 4(a). For details on inserting T-PE into temporal tokens, see Appendix A.2.

V-PE also consists of a geometric PE and a semantic PE. The **geometric PE** employs Depthwise Convolution (DWConv) (Chollet 2017) and is added to the input embedding of each attention layer. The locations where this PE is added are masked by the blue dashed boxes in Figures 4(b). The way to insert the **semantic PE** is the same as T-PE. The locations where the semantic PE is inserted are masked by the purple dashed boxes in Figures 4(c). For details on inserting V-PE into variable tokens, see Appendix A.3.

T2B-PE is a Transformer-based dual-branch framework. The left branch of T2B-PE is used to process temporal tokens, while the right branch is for variable tokens. Except for PE, the structure of each branch is identical to a vanilla Transformer encoder. When the historical series is input into T2B-PE, it is first processed into temporal tokens and variable tokens, which are then handled by the left branch with T-PE and the right branch with V-PE, respectively. Subsequently, T2B-PE merges the features from the dual branches through a gated unit designed explicitly for TSF to make predictions. The overall architecture of T2B-PE is shown in Figure 4. For more details on T2B-PE, please refer to Appendix A.4.

Experiments

In this section, we use experiments to validate the effectiveness of T2B-PE. Specifically, we first introduce the experimental settings. Next, we present the comparative experimental results on six benchmark datasets. Finally, we use visualization experiments to validate the effectiveness of T-PE and V-PE. More results are provided in Appendix H-J.

Experimental settings

This subsection introduces the datasets, baselines, and implementation details.

Datasets We utilize six real-world datasets, including ECL, ETTh2, Traffic, Weather in (Liu et al. 2023), Solar-Energy dataset in (Lai et al. 2018), and PEMS03 in (Liu et al. 2022). The train-validation-test set split protocol is the same as (Liu et al. 2023). For further details, refer to Appendix C.

Baselines We choose ten classic Time Series Forecasting (TSF) methods for comparison, including (i) the transformer-based TSF methods using temporal tokens: *FEDformer* (Zhou et al. 2022), *PatchTST* (Nie et al. 2023) and *Autoformer* (Wu et al. 2021a), (ii) the transformer-based TSF method using variable tokens: *iTransformer* (Liu et al. 2023), and (iii) the transformer-based TSF methods using both temporal and variable tokens: *Crossformer* (Zhang and Yan 2023) and *DSformer* (Yu et al. 2023). Additionally, we introduce

Models	Ours (Ours)		FEDformer (2022)		PatchTST (2023)		iTransformer (2023)		Crossformer (2023)		TimesNet (2023)		DLinear (2023)		DSformer (2023)		Autoformer (2021)		SCINet (2022)		TIDE (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	**0.378**	**0.402**	0.437	0.449	0.387	0.407	0.383	0.407	0.942	0.684	0.414	0.427	0.559	0.515	0.410	0.421	0.450	0.459	0.954	0.723	0.611	0.550
ECL	**0.165**	**0.261**	0.214	0.327	0.216	0.304	0.178	0.270	0.244	0.334	0.192	0.295	0.212	0.300	0.205	0.297	0.227	0.338	0.268	0.365	0.251	0.344
Traffic	**0.414**	**0.282**	0.610	0.376	0.555	0.362	0.428	0.282	0.550	0.304	0.620	0.336	0.625	0.383	0.548	0.373	0.628	0.379	0.804	0.509	0.760	0.473
Weather	**0.243**	**0.273**	0.309	0.360	0.259	0.281	0.258	0.279	0.259	0.315	0.259	0.287	0.265	0.317	0.246	0.274	0.338	0.382	0.292	0.363	0.271	0.320
Solar-Energy	**0.231**	**0.258**	0.291	0.381	0.270	0.307	0.233	0.262	0.641	0.639	0.301	0.319	0.330	0.401	0.301	0.327	0.885	0.711	0.282	0.375	0.347	0.417
PEMS03	**0.107**	**0.215**	0.213	0.327	0.180	0.291	0.113	0.221	0.169	0.281	0.147	0.248	0.278	0.375	0.193	0.297	0.667	0.601	0.114	0.224	0.326	0.419

Table 2: Forecasting results with prediction lengths $S \in \{12, 24, 36, 48\}$ for PEMS03 and $S \in \{96, 192, 336, 720\}$ for others. We fix the lookback length $T = 96$. All the results are averaged from all prediction lengths. Detailed results are in Appendix M.

multiple recently proposed methods for comparison, including *TimesNet* (Wu et al. 2023), *DLinear* (Zeng et al. 2023a), *SCINet* (Liu et al. 2022), and *TIDE* (Das et al. 2023). We do not compare with Seformer (Zeng et al. 2023b) because it has not released its source code.

Implementation Details We follow the standard settings as in (Liu et al. 2023). Data undergo mean-std normalization before being input into the model and inverse mean-std normalization before output. The proposed models are trained using an NVIDIA 4090 GPU. MSE and MAE are employed as evaluation metrics. In our experiments, we configure the number of attention layers, L, to 3 for the Weather dataset and 4 for other datasets. The token dimension D is set to 512. The head number of multi-head attention is set to 8. The batch size is set to 16. The geometric PE of V-PE employs Depthwise Convolution (DWConv) (Chollet 2017) with a kernel size of 3, a stride of 1, and padding of 1. For all experiments, we utilize the Adam optimizer (Kingma and Ba 2015), with i) cosine learning rate decay following linear warm-up or ii) decay learning rate. The first strategy requires two hyperparameters: the warm-up length Len_w and the scaling factor f_s. When training reaches the m-th batch, the learning rate for this batch is:

$$\begin{aligned} lr &= f_{s1} * (D^{-0.5} * min(m^{-0.5}, m * Len_w^{-1.5})) \\ f_{s1} &= (512 * Len_w)^{0.5} * f_s, \end{aligned} \tag{3}$$

where f_{s1} is the adjusted scaling factor. The corresponding Len_w and f_s for each dataset are shown in Appendix D.1. To aid understanding, the learning rate variation curve for the first strategy is plotted in Appendix D.1. The second strategy requires one hyperparameter lr_{init}, which is the initial learning rate for the training. When training reaches the k-th epoch, the learning rate for this batch is:

$$lr = lr_{init} * 0.5^k. \tag{4}$$

The corresponding lr_{init} for each dataset is shown in Appendix D.1. The total number of epochs is 40. The training process will stop early if the validation loss doesn't decrease within three epochs.

Comparative Experimental Results

The comprehensive prediction results are shown in Table 2. The best results are marked in red, and the second-best results are marked in blue. Lower MSE/MAE values indicate more accurate predictions. Compared to other TSF models, our method achieves the best average results on each benchmark dataset. The results demonstrate that our proposed method has superior performance. Additionally, the encoder of our model is the same as the vanilla Transformer, with the only difference being the choice of PE. The experimental results show that SOTA performance can be achieved even with the simplest Transformer encoder structure combined with our designed PE. The results prove the effectiveness of T2B-PE.

Performance Boost with T-PE or V-PE

To further validate the effectiveness of T-PE and V-PE, we applied them to three other Transformer-based TSF methods: i) Transformer, ii) PatchTST, and iii) iTransformer. Specifically, We evaluate these three algorithms on the Weather and ECL datasets to observe the performance improvement after incorporating the proposed PEs. Vanilla Transformer and PatchTST deal with temporal tokens, so only T-PE can be used. iTransformer deals with variable tokens, so only V-PE can be used. The experimental results are shown in Table 3. The results show that the models' performance can be improved after using T-PE or V-PE, indicating the effectiveness of T-PE and V-PE.

Table 3: The performance variation of applying T-PE or V-PE to different models. The prediction lengths are $S \in \{96, 192, 336, 720\}$. We fix the lookback length $T = 96$. All the results are averaged from all prediction lengths. Detailed results are in Appendix G.

MSE/MAE	Dataset	Original	+T-PE or V-PE
Transformer	Weather	0.379/0.432	**0.316/0.364**
	ECL	0.333/0.414	**0.283/0.376**
PatchTST	Weather	0.259/0.281	**0.249/0.274**
	ECL	0.216/0.304	**0.207/0.293**
iTransformer	Weather	0.258/0.279	**0.246/0.275**
	ECL	0.178/0.270	**0.169/0.264**

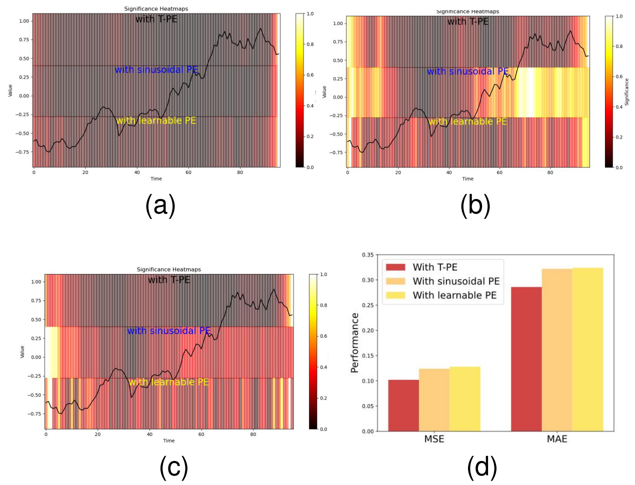


Figure 5: Visualization of model performance and the temporal token significance. The black line represents the input series, and the significance of tokens using the T-PE, the sinusoidal PE, or the learnable PE is depicted in the background. (a) First attention layer; (b) Second attention layer; (c) Third attention layer; (d) Prediction performance.

Visualization Analysis

In this subsection, we construct visualization experiments to evaluate T-PE and V-PE.

To validate the effectiveness of T-PE, we compare the visualized results to see if the model with different PEs can accurately identify positive tokens. When the input series does not contain periodic patterns (Figure 5), two types of input tokens are more likely to become positive tokens: i) the last few tokens and ii) the tokens at peaks or valleys. Therefore, we want to check whether the tokens with high significance in the visualized results mainly belong to these two types. Specifically, we train the Transformer with the sinusoidal PE, the learnable PE, or our T-PE on the Weather dataset. Both the lookback length and prediction length are set to 96. The number of encoder layers is set to 3. We calculate the token significance for each layer and plot it as a heatmap, aligning it with the corresponding regions of the input series (Figure 5). The results demonstrate that, compared to using other PEs, the model with T-PE can accurately identify the last few tokens of the input series as positive at each layer while recognizing most tokens farther from the end of the input series as negative. Identifying these distant tokens as negative is reasonable because when the overall trend of the input sequence is relatively stable, these tokens should have less impact on the prediction. The experimental results validate the effectiveness of T-PE.

For V-PE, we cannot intuitively determine which input token is more important, as we do with temporal tokens. Therefore, we artificially add noise tokens and observe whether the model can distinguish them. Specifically, we train the Transformer with no PE, the sinusoidal PE, the learnable PE, or our V-PE on the Weather dataset. Note that here, the Transformer only takes variable tokens as input. Both the lookback length and prediction length are set to 96. In the Weather dataset, there are 21 variable tokens, and we replace the first 10 variable tokens with Gaussian noise for training. During testing, we similarly replace the first ten tokens of the test data with Gaussian noise, then plot the input token significance as a heatmap and record the predictive performance (as shown in Figure 6). The results show that compared to other PE settings, when the model uses V-PE, the significance of the first ten noisy tokens is well suppressed at each layer, and the predictions yield lower MSE/MAE. The experimental results validate the effectiveness of V-PE.

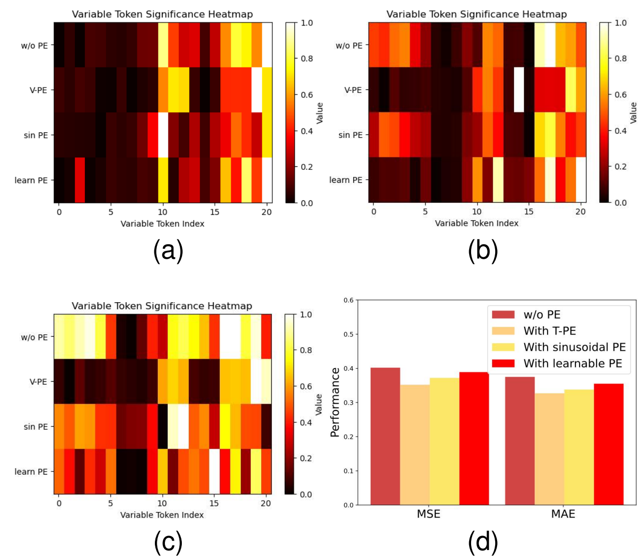


Figure 6: Visualization of model performance and the token significance. (a) First attention layer; (b) Second attention layer; (c) Third attention layer; (d) Prediction performance. Please note that the first ten variable tokens are artificially inserted noise, and the lower their significance, the better.

Conclusion

This paper explores the role of tokens for predictions in Transformer-based TSF methods from both theoretical and experimental perspectives. First, through theoretical derivation, we find that during training, the gradient of the Transformer-based TSF model mainly depends on positive tokens. Then, through a series of experiments, we observe that PE helps the model identify positive tokens. The experiments also reveal that as the network deepens, the information from PE diminishes, which interferes with the model's ability to identify positive tokens. This insight leads us to enhance PE in the deeper layers, and we find this approach alleviates the issue and improves the model's performance. Additionally, experiments show that semantic PE also enhances the model's ability to identify positive tokens and improve the model's performance. Based on these findings, we propose T-PE for temporal tokens and V-PE for variable tokens. To fully utilize T-PE and V-PE, we design a dual-branch framework based on the Transformer, named T2B-PE. We validate the effectiveness of T2B-PE using six real-world datasets and further confirm the effectiveness of T-PE and V-PE through extensive visualization experiments.

References

Abhishek, K.; Singh, M.; Ghosh, S.; and Anand, A. 2012. Weather forecasting model using artificial neural network. *Procedia Technology*, 4: 311–318.

Agrawal, A.; Kumar, V.; Pandey, A.; and Khan, I. 2012. An application of time series analysis for weather forecasting. *International Journal of Engineering Research and Applications*, 2(2): 974–980.

Boussif, O.; Boukachab, G.; Assouline, D.; Massaroli, S.; Yuan, T.; Benabbou, L.; and Bengio, Y. 2024. Improving* day-ahead* Solar Irradiance Time Series Forecasting by Leveraging Spatio-Temporal Context. *Advances in Neural Information Processing Systems*, 36.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Wei, X.; Xia, H.; and Shen, C. 2021. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.

Debnath, K. B.; and Mourshed, M. 2018. Forecasting methods in energy planning models. *Renewable and Sustainable Energy Reviews*, 88: 297–325.

Dong, J.; Wu, H.; Zhang, H.; Zhang, L.; Wang, J.; and Long, M. 2023. SimMTM: A Simple Pre-Training Framework for Masked Time-Series Modeling. *arXiv preprint arXiv:2302.00861*.

Du, D.; Su, B.; and Wei, Z. 2023. Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Fang, Y.; Qin, Y.; Luo, H.; Zhao, F.; and Zheng, K. 2023. STWave+: A Multi-Scale Efficient Spectral Graph Attention Network With Long-Term Trends for Disentangled Traffic Flow Forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Foumani, N. M.; Tan, C. W.; Webb, G. I.; and Salehi, M. 2024. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38(1): 22–48.

Islam, M. A.; Jia, S.; and Bruce, N. D. 2020. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*.

Jiang, K.; Peng, P.; Lian, Y.; and Xu, W. 2022. The encoding method of position embeddings in vision transformer. *Journal of Visual Communication and Image Representation*, 89: 103664.

Karevan, Z.; and Suykens, J. A. 2020. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125: 1–9.

Ke, G.; He, D.; and Liu, T.-Y. 2020. Rethinking positional encoding in language pre-training. *arXiv preprint arXiv:2006.15595*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. *ICLR*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. *SIGIR*.

Li, R.; Zhang, F.; Li, T.; Zhang, N.; and Zhang, T. 2022. DMGAN: Dynamic multi-hop graph attention network for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Liang, D.; Zhang, H.; Yuan, D.; Ma, X.; Li, D.; and Zhang, M. 2023. Does Long-Term Series Forecasting Need Complex Attention and Extra Long Inputs? *arXiv preprint arXiv:2306.05035*.

Lim, B.; Arık, S. Ö.; Loeff, N.; and Pfister, T. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *ICLR*.

Novo, R.; Marocco, P.; Giorgi, G.; Lanzini, A.; Santarelli, M.; and Mattiazzo, G. 2022. Planning the decarbonisation of energy systems: The importance of applying time series clustering to long-term models. *Energy Conversion and Management: X*, 15: 100274.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Steuer, R.; Kurths, J.; Daub, C. O.; Weise, J.; and Selbig, J. 2002. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl_2): S231–S240.

Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; and Liu, Y. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063.

Tang, P.; and Zhang, X. 2023. Infomaxformer: Maximum Entropy Transformer for Long Time-Series Forecasting Problem. *arXiv preprint arXiv:2301.01772*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *NeurIPS*.

Wang, H.; Chen, J.; Fan, Z.; Zhang, Z.; Cai, Z.; and Song, X. 2022. St-expertnet: A deep expert framework for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. *ICLR*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021a. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. *NeurIPS*.

Wu, K.; Peng, H.; Chen, M.; Fu, J.; and Chao, H. 2021b. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10033–10041.

Wu, N.; Green, B.; Ben, X.; and O'Banion, S. 2020. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.

Yu, C.; Wang, F.; Shao, Z.; Sun, T.; Wu, L.; and Xu, Y. 2023. DSformer: a double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3062–3072.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023a. Are Transformers Effective for Time Series Forecasting? *AAAI*.

Zeng, P.; Hu, G.; Zhou, X.; Li, S.; and Liu, P. 2023b. Seformer: a long sequence time-series forecasting model based on binary position encoding and information transfer regularization. *Applied Intelligence*, 53(12): 15747–15771.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. *ICLR*.

Zhao, Y.; Ma, Z.; Zhou, T.; Ye, M.; Sun, L.; and Qian, Y. 2023. GCformer: An Efficient Solution for Accurate and Scalable Long-Term Multivariate Time Series Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3464–3473.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *ICML*.