AVGNets: Angular Visibility Graph Networks with Probability Attention for Time Series Forecasting

Shengzhong Mao , Xiao-Jun Zeng
Department of Computer Science, University of Manchester
{shengzhong.mao, x.zeng}@manchester.ac.uk

Abstract

Visibility algorithm is a mapping that bridges network representation learning of time series, which has been broadly investigated for forecasting tasks. Due to the mechanism of visibility transformation, however, the constructed graph is solely structured by a binary adjacency matrix, which inevitably leads to sequence information loss during the mapping. To this end, we propose Angular Visibility Graph Networks (AVGNets), which are designed with the following major features: (i) the framework reconstructs weighted graphs for time series encoding concerning topological structure of visual angle in visibility networks, which succeeds sequential and structural characteristics described by weighted angular matrix. (ii) *ProbAttention* module is newly introduced for evaluating probabilistic attention distribution of weighted networks, with inspiring capacities to capture inner and outer temporal correlation across multi-layer graphs. Extensive experiments and ablation studies on real-world datasets covering multiple ranges show that AVGNets outperform existing state-of-the-art forecasting models, providing a novel perspective of graph representation for sequence modeling.

1 Introduction

Time series forecasting, cast as a classical supervised machine learning problem, is to utilize past observations to predict future patterns where temporal component is involved. It has wide-ranging applications in the domain of energy [Zeng *et al.*, 2021], business [Jin *et al.*, 2022], and more [Stankeviciute *et al.*, 2021; Chauhan *et al.*, 2022]. Given its broad practical usage and great impact on modern society, forecasting has drawn considerable concentration from academia to industry. However, with the temporal order dependence between observations, which is the inherent time series structure and also the constraint to forecast, complicated mechanism of sequential variations and dependent variables make it less predictable. Therefore, time series forecasting has always been a heated while challenging task since it appears.

Typically, existing time series forecasting models can be classified into two categories [Liu *et al.*, 2021]: the classical statistical models and modern deep learning methods. Classical models, based on statistical theories, characterize time series using random variables to simulate time series predictions. Among classical approaches, autoregressive integrated moving average (ARIMA) [Box *et al.*, 2015] and generalized autoregressive conditional heteroskedasticity (GARCH) [Bollerslev, 1986] are considered as representative models. Along with improved computing resources, deep learning based models, as the group of modern methods, emerge to be more powerful predictors in handling nonlinear patterns, represented by recurrent neural network (RNN) [Zhao *et al.*, 2020; Salinas *et al.*, 2020] and its variant long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997]. Recently, transformer [Vaswani *et al.*, 2017] model based on attention mechanism has been successfully applied to time series prediction. Lately more efficient transformer-based structures are presented [Zhou *et al.*, 2021; Wu *et al.*, 2021; Zhou *et al.*, 2022], further improving predictive performance with less memory use and time consumption.

Past decades witnessed the successful application of both classical and modern models in forecasting time series. Nevertheless, their limitations also exist. Regarding classical models, they are a set of competitive strategies for linear predictions, while lacking the ability to capture nonlinearity and nonstationarity results in decreasing predictive accuracy facing volatile time series [De Gooijer and Hyndman, 2006]. Furthermore, it requires satisfying statistical assumptions and manually modeling time series features, for example, tendency and seasonality. As for deep learning mechanism, like LSTM, it has better ability modeling nonlinear features, while inevitably leads to the vanishing gradient and complicated dependency between input and output [Lin *et al.*, 2021; Lim and Zohren, 2021]. Moreover, poor interpretability limits the practical application, especially when observed data size is insufficient, the predictive power cannot be guaranteed, easily to causing overfitting and slow convergence.

Due to the deficiencies of existing methods in processing dynamic sequences and modeling temporal dependencies, there is a need to develop alternative approaches and explore time series uncertainties different from traditional

The 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)@AI4TS: AI for Time Series Analysis.

views. Among these options, the visibility algorithm, as a tool for mapping time series into networks, has been demonstrated to be a new possibility and direction for the competence of sequential feature extraction by graph structure [Lacasa *et al.*, 2008; Iacovacci and Lacasa, 2016; Zou *et al.*, 2019]. The core concept is to convert time series into visibility network by visible criterion, where data are thereby mapped as nodes in a graph [Lacasa *et al.*, 2008]. Since the proposal of visibility algorithm, a variety of models and variants are developed for time series analysis [Xuan *et al.*, 2022; Mao *et al.*, 2021; Zeng and Tang, 2022], and are also extended to other fields like image classification [Iacovacci and Lacasa, 2019] and signal processing [Olamat *et al.*, 2022]. However, the weakness of visibility transformation is also obvious, since the observed data of time series are simply mapped into binary visible relations during the transformation process, and visibility network is consequently only structured by an adjacency matrix, which inevitably leads to information loss of sequential values. To tackle these issues, the angular visibility graph encoding strategy is proposed, as a novel weighted graph representation of time series. Then we introduce a novel *ProbAttention* module capturing long-term dependencies of intra and inter-layers to further develop Angular Visibility Graph Networks (AVGNets). The main contributions are summarized below.

- The framework reconstructs weighted graphs for encoding time series concerning topology structure of visual angle in visibility networks, which succeeds sequential characteristics described by weighted angular matrix.
- *ProbAttention* module is newly introduced for evaluating probabilistic attention score of weighted networks, with inspiring capacities to capture inner and outer temporal correlation across multi-layer graphs.
- The accuracy of forecasting was improved on the real-world datasets. Experiments demonstrates predictive power of proposed architecture, providing a bridge for forecasting tasks from a fresh view of graph networks.

2 Related Works

In this section, we give a more detailed description of time series forecasting methods including two types: attention-based models and graph-based models.

2.1 Attention-based Forecasting Models

Since the success of vanilla Transformer [Vaswani *et al.*, 2017] in natural language processing, attention mechanism attracts heated focus for handling time series forecasting problem, as one classic sequential processing task. LogTrans [Li *et al.*, 2019] introduced convolutional self-attention mechanism to produce queries and keys by causal convolution, which incorporated local context into standard attention for enhancing model predictability. Reformer [Kitaev *et al.*, 2020] developed locally sensitive hashing (LSH) attention strategy to approximate attention by allocating similar query. By extending self-attention with KL-divergence criterion for selecting primary queries, Informer [Zhou *et al.*, 2021] was consequently proposed for capturing long-term dependencies between the outputs and inputs. The introduced *ProbSparse* attention mechanism reduced time complexity to $O(LlogL)$. Autoformer [Wu *et al.*, 2021] is another popular model that designed an auto-correlation module to substitute vanilla attention block by devising a seasonal trend decomposition architecture. FEDformer [Zhou *et al.*, 2022] achieved great success by combining Fourier transform with attention mechanism in the frequency domain. Since this inspiring work, applying self-attention mechanism in frequency domain into modeling time series has been worth exploring.

2.2 Graph-based Forecasting Models

Graph model is another category of competitive technique for modeling time series, which maps time series into networks and investigates sequence based on transformed graph structure. Based on the visibility criterion, visibility graph [Lacasa *et al.*, 2008; Lacasa and Iacovacci, 2017] was proposed to convert time series into visibility networks where visible connectivity was summarized by an adjacency matrix. Since the proposal of visibility algorithm, it has been widely adopted in time series analysis and other domains [Iacovacci and Lacasa, 2019; Olamat *et al.*, 2022; Mao and Zeng, 2023]. Afterward, to deal with multivariate time series, multiplex visibility graph [Lacasa *et al.*, 2015] was developed. With the combination of graph theory and neural networks, the graph neural networks (GNN) was developed. MTGNN [Wu *et al.*, 2020] proposed a general graph neural network framework designed specifically for multivariate time series data, which extracts the uni-directed relations among variables through a graph learning module, into which external knowledge like variable attributes can be easily integrated. Z-GCNets [Chen *et al.*, 2021a] introduced the concept of zigzag persistence into time-aware graph convolutional networks to enhance deep learning architectures with the most salient time-conditioned topological information of the data. S2GCNets [Chen *et al.*, 2021b] integrated time-aware deep learning and multi-persistence, and constructed a convolution module that simultaneously accounts for the extracted spatio-temporal dependencies.

3 Methodology

In this section, we will describe how angular visibility graph works to convert time series into networks, and introduce *ProbAttention* mechanism as a module of proposed structure.

3.1 Problem Definition

Firstly we define the forecasting task in this work. Suppose the input time series $\mathcal{X}^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_{L_x}^t | x_i^t \in \mathbb{R}^{d_x}\}$ at time step t with window size L_x, time series forecasting is to output predict set $\mathcal{Y}^t = \{\mathbf{y}_1^t, ..., \mathbf{y}_{L_y}^t | y_i^t \in \mathbb{R}^{d_y}\}$ where L_y is the forecast length. This work introduces AVGNets applicable for both univariate ($d_y = 1$) and multivariate ($d_y > 1$) forecasting.

3.2 Angular Visibility Graph

Visibility algorithm [Lacasa *et al.*, 2008] is a technique that transforms time series into networks by intuitively visible relation. Given a time series set $\mathbf{x} = \{x_i\}_{i=1}^t$, the core of visibility criterion can be formally defined as

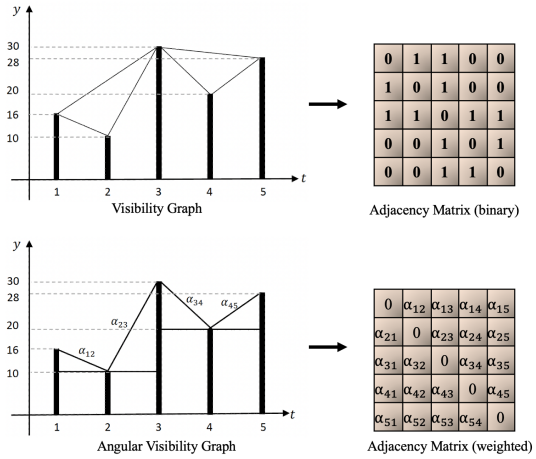

Figure 1: The process of transformation from time series to angular visibility graph. By introducing the angle $\alpha_{ij}$to measure angular relationship between any two vertices, the angular visibility graph is constructed where the structure is represented by weighted adjacency matrix $\mathbf{A}_{\mathbf{w}}^{\mathbf{n}\times\mathbf{n}} = \{\tan\alpha_{ij}\}_{i,j=1}^{n}$.

Definition 1. *A pair of time series data (x_i, x_j) has visibility, and will be linked as two nodes in the visibility network, if the criteria [Lacasa* et al.*, 2008]*

$$x_k < x_j + \frac{j-k}{j-i}(x_i - x_j), \tag{1}$$

is satisfied for any x_k with $i < k < j$, where x is observed value and k denotes temporal order.

According to the definition of visibility algorithm, a time series can be encoded into a network and represented by graph $G(V, E)$, where data values are encoded as vertex set $V = \{v_i\}_{i=1}^{n}$, and $E = \{e_{ij}\}_{i,j=1}^{n}$. The generated graph is structured utilizing visibility criteria, which can be presented by adjacency matrix $\mathbf{A}^{n\times n}$, written as

$$\mathbf{A}^{n\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

where a_{ij} represents the visibility of two vertices. According to the definition of visibility algorithm, the value of a_{ij} is binary where $a_{ij} = 1$ if $e_{ij} \in E$ otherwise $a_{ij} = 0$.

According to the adjacency matrix, however, as visibility algorithm converts time series data into nodes in the graph, and the visibility between them is only visible or not, i.e. the binary relationship, the information loss of sequential data values inevitably occurs as constructed network only determined by a binary adjacency matrix, failing to track underlying variant structures among time series data streams. This is determined by the inherent mapping mechanism itself and few works have addressed it. Therefore, the angular visibility graph is presented to cope with these issues.

For illustrative purpose, we present angular visibility transformation process in Fig. 1. In the upper part, a time series containing five nodes is converted into visibility graph and represented by binary adjacency matrix, while in the lower zone an angle α_{ij} is introduced to measure angular relationship between any two vertices v_i and v_j, where $\tan\alpha_{ij} = (x_j - x_i)/(j - i)$. Consequently, the weighted adjacency matrix is defined by $\mathbf{A}_{\mathbf{w}}^{\mathbf{n}\times\mathbf{n}} = \{\tan\alpha_{ij}\}_{i,j=1}^{n}$. By applying the above operation into multivariate time series, we can obtain multiplex visibility graph [Lacasa *et al.*, 2015] as shown in Fig. 2, and angular graphs do so into multi-layer networks.

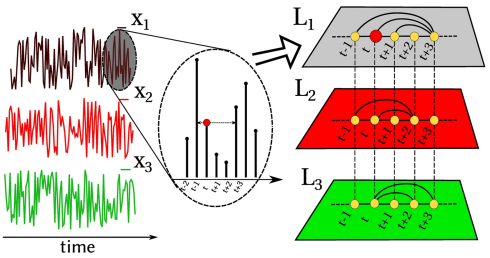

Figure 2: From multivariate time series into multiplex visibility graph [Lacasa *et al.*, 2015]. For each time series $\mathbf{x_i}$, it can be mapped into one layer network $\mathbf{L_i}$ in multiplex visibility graph.

3.3 *ProbAttention* Mechanism

Since temporal sequences are converted into graphs, for forecasting purpose, a probability score is presented to measure the possibility that a link exists between a pair of nodes in a network [Liu and Lü, 2010]. The probability is initialized by transition probability matrix $\mathbf{P}$, where $P_{ij} = a_{ij}/d_i$. Here a_{ij} is the element of adjacency matrix and d_i refers to the degree of node. After iterating t steps, the transition probability denoted by $\boldsymbol{\pi}(t)$ is defined as

$$\boldsymbol{\pi}_i(t) = P^T \boldsymbol{\pi}_i(t-1), \tag{2}$$

where the initialization state is denoted as $\boldsymbol{\pi}_i(0)$ whose i-th item is 1 and others equal to 0. t is sufficiently large to allow $\pi_i(t) - \pi_i(t-1) < \epsilon$ and generally $\epsilon < 10^{-5}$, making sure all nodes are randomly visited. Successively, the probability score would be determined by [Liu and Lü, 2010]

$$ProbScore_{ij}(t) = \frac{d_i}{2N}\pi_{ij}(t) + \frac{d_j}{2N}\pi_{ji}(t), \tag{3}$$

where π_{ij} refers to the j-th element of $\boldsymbol{\pi}_i$, and N denotes the number of edges. The final $ProbScore$ is the superposition of $ProbScore(t)$ at each time step t.

Given the weighted angular graph $G^w(V^w, E^w)$, the degree of node is denoted by $d_i^w = \sum_{j=1}^{n} a_{ij}^w$, which implies the higher degree, the greater importance of node. Besides, in graph theory the correlation, or similarity, of two nodes is inversely proportional to the distance between them, which means the farther apart two nodes are, the less correlated they are [Liu and Lü, 2010]. To characterize variable correlations in a graph structure, therefore, we give a modified probability distribution of any two nodes considering network features, defined by $Prob_{ij} = \mathbf{P}_{ij}^m (d_i^w d_j^w)/l_{ij}^2$, where l_{ij} denotes the distance of vertex v_i^w and v_j^w. $\mathbf{P}^m$ is the probability matrix, written as

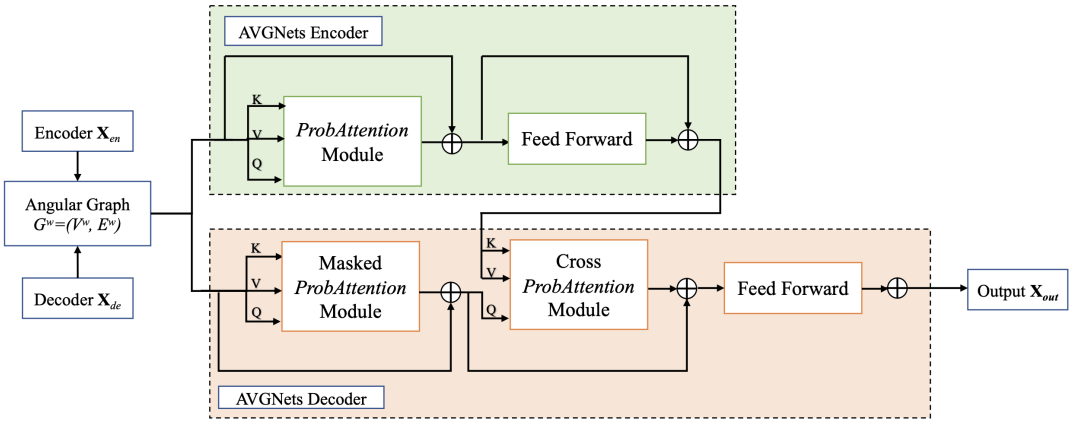

Figure 3: The flowchart of proposed framework consisting of encoder and decoder parts. For encoder part, the input time series will be transformed into angular visibility graph and fed into *ProbAttention* module to measure the similarity of $\mathbf{Q}$ and $\mathbf{K}$. In the decoder part, likewise, decoder series do the same transformation, and input the graph structure into masked *ProbAttention* module. Then a crossed *ProbAttention* is adopted to determine the attention scores.

$$\mathbf{P}^m_{n\times n} = \begin{bmatrix} p_{v_1v_1} & p_{v_1v_2} & \cdots & p_{v_1v_n} \\ p_{v_2v_1} & p_{v_2v_2} & \cdots & p_{v_2v_n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{v_nv_1} & p_{v_nv_2} & \cdots & p_{v_nv_n} \end{bmatrix}$$

where $p_{v_iv_j} = ProbScore_{ij}$ is calculated by Eq. (3).

Typical attention mechanism [Vaswani *et al.*, 2017] is a mapping function that inputs queries, keys and values, and outputs weighted summation of input values. Consider $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the input vector of queries, keys and values, the attention function is mathematically defined as $Attention(\mathbf{Q},\mathbf{K},\mathbf{V}) = softmax(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})\mathbf{V}$, where d_k is the dimension of $\mathbf{K}$. In our model, we design a novel module called *ProbAttention* mechanism based on the probability distribution of angular visibility network. For input matrix $\mathbf{X} \in \mathbb{R}^{L_x\times d}$ where L_x is the length and d is the dimension of sequence, the projection matrices are determined by $\mathbf{Q}^{(i)} = \mathbf{W}_q^T\mathbf{X}^{(i)}$, $\mathbf{K}^{(i)} = \mathbf{W}_k^T\mathbf{X}^{(i)}$, $\mathbf{V}^{(i)} = \mathbf{W}_v^T\mathbf{X}^{(i)}$, where $\mathbf{W}_q \in \mathbb{R}^{L_q\times d}$, $\mathbf{W}_k \in \mathbb{R}^{L_k\times d}$, and $\mathbf{W}_v \in \mathbb{R}^{L_v\times d}$ are the learned parameter matrix. The *ProbAttention* is thereby defined by

$$ProbAttention(\mathbf{Q},\mathbf{K},\mathbf{V}) = \sigma(Prob(\mathbf{Q},\mathbf{K}))\mathbf{V}, \quad (4)$$

$$Prob(\mathbf{Q^{(i)}},\mathbf{K}) = \exp(\sum_j ProbScore(q_i,k_j)\frac{d_{q_i}d_{k_j}}{l^2_{q_ik_j}}), \quad (5)$$

$$ProbScore(q_i,k_j) = \sum_{t=1}^{n}[\phi_{q_i}\pi_t(q_i,k_j)+\phi_{k_j}\pi_t(k_j,q_i)], \quad (6)$$

where q_i and k_j are the i-th and j-th row of $\mathbf{Q}$ and $\mathbf{K}$, respectively. σ is the $softmax$ activation function, and $\exp(\cdot)$ is the exponential function to magnify the item with a higher probability. $\phi_i = d_i/|E|$ and E is the edge set.

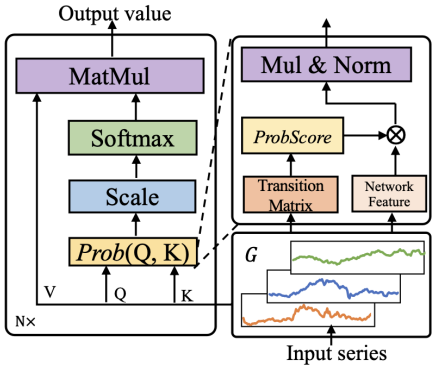

Figure 4: *ProbAttention* module. The block takes sequential input and transforms time series into angular visibility graph, where each dimension of input time series matrix would be encoded into weighted adjacency matrix. The projection layer generates sequential matrices and then performs probability attention computing $Prob(Q,K)$ considering $ProbScore$ and network features. The scores are assigned as weight coefficients to the output of $\mathbf{V}$.

3.4 Forecasting Framework

The framework incorporates standard encoder-decoder architecture as vanilla transformer [Vaswani *et al.*, 2017] to make predictions. As shown in the flowchart of Fig. 3, following the transformation from time series into angular visibility graph, each dimension of input time series matrix would be encoded into weighted adjacency matrix $\mathbf{A_w}$. Then the transformation layer projects $\mathbf{A_w}$ into representative matrices $\{\mathbf{Q},\mathbf{K},\mathbf{V}\}$, which are fed into *ProbAttention* module to measure the similarity of $\mathbf{Q}$ and $\mathbf{K}$. In the attention layer, the block computes probability attention score $Prob(\mathbf{Q},\mathbf{K})$, and assigns score as weight coefficient to $\mathbf{V}$. Different from classic attention using dot product to calculate scores, *ProbAttention* considers both modified probability distribution and network features to output a weighted sum of $\mathbf{V}$. Fig. 4 is provided to magnify details of how *ProbAttention* connects

Table 1: Results of multivariate forecasting on six datasets measuring by MSE/MAE. The length of input is fixed as L_x = 96 by default, and forecasting horizon $L_y \in \{96, 192, 336, 720\}$. The best results are highlighted in **bold**.

Type		Ours				Attention-based						Graph-based					
Model		AVGNets-e		AVGNets-d		FEDformer		Autoformer		Informer		S2GCNets		Z-GCNets		MTGNN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh_1$	96	**0.365**	**0.393**	0.370	0.402	0.376	0.419	0.449	0.459	0.865	0.713	0.735	0.861	0.724	0.715	0.713	0.910
	192	**0.374**	**0.411**	0.409	0.423	0.420	0.448	0.500	0.482	1.008	0.792	0.813	0.888	0.823	0.874	0.933	0.962
	336	**0.412**	**0.424**	0.448	0.456	0.459	0.465	0.521	0.496	1.107	0.809	0.891	1.002	0.943	0.952	1.044	1.002
	720	**0.496**	**0.489**	0.502	0.497	0.506	0.507	0.514	0.512	1.181	0.865	0.984	1.254	1.285	1.208	1.656	1.049
$ETTh_2$	96	**0.288**	**0.324**	0.304	0.331	0.346	0.388	0.358	0.397	3.755	1.525	0.603	0.630	0.511	0.706	0.822	0.828
	192	**0.343**	**0.376**	0.362	0.398	0.429	0.439	0.456	0.452	5.602	1.931	0.719	0.795	0.585	0.927	0.973	0.938
	336	**0.398**	**0.394**	0.423	0.412	0.496	0.487	0.482	0.486	4.721	1.835	0.841	0.960	0.735	1.181	1.088	1.017
	720	**0.437**	**0.432**	0.443	0.441	0.463	0.474	0.515	0.511	3.647	1.625	0.881	1.077	1.816	1.263	1.517	1.020
Electricity	96	**0.158**	**0.241**	0.161	0.253	0.193	0.308	0.201	0.317	0.274	0.368	0.317	0.441	0.295	0.456	0.446	0.646
	192	0.171	**0.256**	**0.169**	0.261	0.201	0.315	0.222	0.334	0.296	0.386	0.333	0.452	0.369	0.574	0.650	0.668
	336	**0.188**	**0.284**	0.183	0.289	0.214	0.329	0.231	0.338	0.300	0.394	0.361	0.712	0.431	0.892	0.961	0.681
	720	0.198	**0.293**	**0.196**	0.313	0.246	0.355	0.254	0.361	0.373	0.439	0.397	0.736	0.642	0.954	0.975	0.725
Traffic	96	0.418	**0.254**	**0.403**	0.284	0.587	0.366	0.613	0.388	0.719	0.391	0.841	0.929	0.554	0.660	0.879	0.979
	192	**0.435**	**0.297**	0.448	0.302	0.604	0.373	0.616	0.382	0.696	0.379	0.891	1.071	0.567	0.745	0.987	0.985
	336	**0.452**	**0.321**	0.469	0.332	0.621	0.383	0.622	0.337	0.777	0.420	0.933	1.122	0.602	0.864	1.095	0.951
	720	**0.467**	0.358	0.471	**0.354**	0.626	0.382	0.660	0.408	0.864	0.472	0.937	1.267	0.894	0.925	1.647	1.025
Weather	96	**0.163**	**0.234**	0.169	0.239	0.217	0.296	0.266	0.336	0.300	0.384	0.333	0.455	0.314	0.817	0.454	0.681
	192	**0.198**	**0.287**	0.202	0.293	0.276	0.336	0.307	0.367	0.598	0.544	0.399	0.536	0.430	0.980	0.692	0.751
	336	**0.264**	**0.313**	0.289	0.342	0.339	0.380	0.359	0.395	0.578	0.523	0.573	0.658	0.543	1.040	0.738	0.821
	720	0.353	0.379	**0.345**	**0.374**	0.403	0.428	0.419	0.428	1.059	0.741	0.725	0.995	0.617	1.107	0.894	0.897
Exchange	96	**0.102**	**0.198**	0.123	0.218	0.148	0.278	0.197	0.323	0.847	0.752	0.379	0.872	0.405	0.822	0.932	0.622
	192	**0.128**	**0.237**	0.153	0.241	0.271	0.380	0.300	0.369	1.204	0.895	0.653	0.915	0.639	0.974	1.134	0.750
	336	**0.298**	**0.312**	0.328	0.321	0.460	0.502	0.509	0.524	1.672	1.036	1.274	1.166	1.089	1.158	1.718	1.031
	720	**0.662**	**0.487**	0.853	0.512	1.195	0.841	1.447	0.941	2.478	1.310	1.875	1.652	1.388	1.392	1.981	1.938

angular graph and takes $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\}$ as input to compute attention score. Besides, the encoder block includes norm layers and a feed forward network with residual connections as shown in Fig. 3.

The decoder part consists of stacked six identical layers. Firstly, decoder block does the same transformation, and inputs the graph structure into masked *ProbAttention* module. Here masked operation is to set masked dot-products to infinite in case of data leakage, making sure that the predictions can depend only on the known outputs at previous positions. Then a crossed *ProbAttention* is adopted to perform multi-head attention over the output of the encoder stack, to determine the attention scores. Likewise, each of the sub-layers are with residual connections followed by layer normalization to obtain the prediction results. AVGNets apply to univariate and multivariate time series foresting, depending on whether the angular visibility graph is single or multilayer.

4 Experiments

4.1 Datasets

Experiments are conducted on several benchmarks, covering forecasting applications in the domain of energy, traffic, climate, economy, etc. Datasets are split into train, validation, and test set following the setting as 6:2:2 for ETT and 7:1:2 for others [Zhou *et al.*, 2022; Wu *et al.*, 2021].

ETT (Electricity Transformer Temperature) contains the oil temperature of electricity transformer reflecting the capacity of extreme load. $ETTh_1$ and $ETTh_2$ are two variants of datasets recorded hourly from July 2016 to July 2018.

Electricity includes electric power consumption data collected from UCI machine learning repository, which is sampled every hour of 321 clients from 2012 to 2014.

Traffic describes the rate of occupied freeways in San Francisco, which is measured every hour by 862 sensors ranging from January 2015 to December 2016.

Weather contains meteorological data of 21 weather indicators in 10-minute level from the Weather Station of the Max Planck Biogeochemistry Institute in the year of 2020.

Exchange covers the daily exchange rates in eight countries during the year from 1990 to 2016.

4.2 Baselines

For multivariate forecasting, since predictive performances of traditional statistical-based models (like ARIMA [Box *et al.*, 2015], DeepAR [Salinas *et al.*, 2020]) and variant RNN models (like LSTM [Hochreiter and Schmidhuber, 1997]) are not satisfying compared with Transformer-type models as demonstrated in [Zhou *et al.*, 2021; Wu *et al.*, 2021], to make comprehensive comparison, we mainly select state-of-the-art (i) attention-based models: FEDformer [Zhou *et al.*, 2022],

https://github.com/zhouhaoyi/ETDataset

https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams
http://pems.dot.ca.gov
https://www.bgc-jena.mpg.de/wetter/
https://github.com/laiguokun/multivariate-time-series-data

Table 2: Results of univariate forecasting on full ETT datasets. $ETTh_1$/$ETTh_2$ are recorded hourly, and $ETTm_1$/$ETTm_2$ are recorded every fifteen minutes. The best results are highlighted in **bold**, indicating AVGNets achieves state-of-the-art performance.

Type		Ours				Attention-based						Graph-based				Statistical			
Model		AVGNets-e		AVGNets-d		FEDformer-f		FEDformer-w		Autoformer		S2GCNets		Z-GCNets		ARIMA		DeepAR	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh_1$	96	**0.062**	**0.171**	0.068	0.232	0.079	0.215	0.080	0.214	0.071	0.206	0.184	0.367	0.281	0.483	0.543	0.571	0.356	0.531
	192	**0.073**	**0.200**	0.074	0.199	0.104	0.245	0.105	0.256	0.114	0.262	0.223	0.375	0.241	0.423	0.564	0.594	0.360	0.520
	336	**0.085**	**0.230**	0.091	0.228	0.119	0.270	0.120	0.269	0.107	0.258	0.213	0.379	0.394	0.543	0.624	0.594	0.428	0.565
	720	**0.124**	**0.286**	0.148	0.299	0.142	0.299	0.127	0.280	0.126	0.283	0.193	0.352	0.443	0.631	0.758	0.683	0.493	0.610
$ETTh_2$	96	**0.113**	0.263	0.132	**0.262**	0.128	0.271	0.156	0.306	0.153	0.306	0.221	0.394	0.212	0.386	0.983	0.938	0.634	0.590
	192	**0.171**	**0.302**	0.175	0.318	0.185	0.330	0.238	0.380	0.204	0.351	0.213	0.392	0.274	0.432	2.732	1.633	1.937	0.889
	336	**0.201**	**0.354**	0.206	0.353	0.231	0.378	0.271	0.412	0.246	0.389	0.232	0.412	0.294	0.432	4.547	1.245	1.791	0.867
	720	0.268	**0.395**	**0.256**	0.403	0.278	0.420	0.288	0.438	0.268	0.409	0.243	0.414	0.229	0.387	2.135	1.134	0.870	0.682
$ETTm_1$	96	**0.027**	**0.120**	0.028	0.119	0.033	0.140	0.036	0.149	0.056	0.183	0.128	0.289	0.064	0.112	0.287	0.343	0.171	0.328
	192	**0.041**	**0.149**	0.043	0.147	0.058	0.186	0.069	0.206	0.081	0.216	0.149	0.302	0.165	0.320	0.433	0.482	0.266	0.427
	336	**0.055**	0.173	0.058	**0.171**	0.084	0.231	0.071	0.209	0.076	0.218	0.403	0.514	0.292	0.461	0.545	0.565	0.454	0.604
	720	**0.076**	**0.202**	0.079	0.204	0.102	0.250	0.105	0.248	0.110	0.267	0.425	0.579	0.421	0.586	0.746	0.729	0.589	0.677
$ETTm_2$	96	0.065	**0.169**	**0.063**	0.179	0.067	0.198	0.063	0.189	0.065	0.189	0.091	0.231	0.078	0.219	0.068	0.224	0.138	0.276
	192	**0.091**	**0.220**	0.093	0.219	0.102	0.245	0.110	0.252	0.118	0.256	0.129	0.274	0.133	0.266	0.137	0.286	0.151	0.343
	336	**0.117**	**0.253**	0.120	0.255	0.130	0.279	0.147	0.301	0.154	0.305	0.145	0.346	0.175	0.312	0.148	0.356	0.185	0.377
	720	**0.171**	**0.310**	0.174	0.313	0.178	0.325	0.219	0.368	0.182	0.335	0.312	0.443	0.158	0.323	0.156	0.344	0.229	0.424

Autoformer [Wu *et al.*, 2021], Informer [Zhou *et al.*, 2021], and representative (ii) graph-based models: S2GCNets [Chen *et al.*, 2021b], Z-GCNets [Chen *et al.*, 2021a], and MTGNN [Wu *et al.*, 2020].

For univariate forecasting, more competitive models are included for comparison. Considering the outstanding capacity of statistical models in capturing sequential features, like trend and seasonality advantageous to predict time series, two representative models of DeepAR [Salinas *et al.*, 2020] and ARIMA [Box *et al.*, 2015] are included. Since FEDformer variants (-f and -w) perform the best on different datasets for univariate tasks, we included both of them, i.e., FEDformer-f and FEDformer-w for full comparisons. Besides, we propose two variants of AVGNets, called AVGNets-e and AVGNets-d, which consider network features of edge strength and degree of node defined in multiplex visibility graph [Lacasa *et al.*, 2015] in computing $ProbScore$, respectively.

4.3 Implementation

Experiments are implemented by Pytorch [Paszke *et al.*, 2019] and conducted on NVIDIA RTX 3090 24GB GPU. Adam [Kingma and Ba, 2014] is selected as model optimizer using L2 loss with learning rate initialized as $1e^{-4}$. The batch size is set as 32 and training process is early stopped within 10 epochs. Experiments are repeated five times alleviating random errors and averaged performances are reported. For comparison purposes, the length of input is fixed as L_x = 96 by default, and forecasting horizon $L_y \in \{96, 192, 336, 720\}$. MSE = $\frac{1}{n}\sum_{i=1}^{n}(\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i})^2$ and MAE = $\frac{1}{n}\sum_{i=1}^{n}|\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}|$ are adopted as evaluating metrics, where the lower the better.

4.4 Main Results

Table 1 lists multivariate forecasting results, from which it can be observed that: (i) the proposed models improve predictive performances on all datasets in terms of two metrics, compared with both attention-based and graph-based models. Additionally, with the growth of forecasting length, AVGNets increase a slower error, which shows the predictive stability of proposed models. (ii) AVGNets-e outperforms AVGNets-d on most datasets, while performing less well in some cases when forecasting electricity, traffic and weather. The winning of AVGNets-d counts 4 in terms of MSE and counts 2 in MAE, which indicates both edge and node features contribute to the forecasting, motivating our further research on graph structure study. (iii) Since FEDfomer is the best baseline model, we make a specific comparison with it. Consider the best results of proposed models and the results of FEDformer, the average improvement of AVGNets is by 6.5%, 15.5%, 19.6%, 27.8%, 22.5%, 40.9% in MSE on six datasets orderly, and by 6.7%, 11.5%, 17.9%, 18.6%, 16.4%, 36.5% in MAE. It can be found that AVGNets show major percentage of improvement in Exchange forecasting which is with great volatility and non-stationarity, indicating AVGNets have a competitive capacity to deal with non-stationary time series.

As for univariate forecasting, following the same experimental setting as in [Zhou *et al.*, 2022], we present results

Table 3: Ablation study for exploring the contribution of angular weighted networks and *ProbAttention* module. ELEC and EXCH refer to the abbreviations of electricity and exchange.

Model	Metric	$ETTh_1$	$ETTh_2$	ELEC	Traffic	Weather	EXCH
Informer	MSE	1.040	4.431	0.311	0.764	0.634	1.550
	MAE	0.795	1.729	0.397	0.416	0.548	0.998
Autoformer	MSE	0.496	0.453	0.227	0.628	0.338	0.613
	MAE	0.487	0.462	0.338	0.379	0.382	0.539
FEDformer	MSE	0.440	0.434	0.214	0.610	0.309	0.519
	MAE	0.460	0.447	0.327	0.376	0.360	0.500
AVGNets/wp	MSE	0.412	0.367	0.172	0.441	0.242	0.298
	MAE	0.429	0.382	0.289	0.307	0.302	0.309
AVGNets/w	MSE	0.385	0.354	0.163	0.423	0.214	0.267
	MAE	0.363	0.351	0.262	0.258	0.288	0.263
AVGNets/p	MSE	0.391	0.363	0.171	0.438	0.226	0.284
	MAE	0.395	0.362	0.279	0.269	0.393	0.286
AVGNets	MSE	0.332	0.276	0.132	0.344	0.162	0.212
	MAE	0.309	0.317	0.221	0.207	0.252	0.230

Table 4: Comparison *ProbAttention* module with other variants self-attention mechanisms, including *ProbAttention*, *LogSparse*, Auto-Correlation and Local-Sensitive Hashing (LSH) Attentions, by replacing *ProbAttention* in AVGNets with them. Experiments are conducted on ETT dataset. "–" indicates no results due to out-of-memory.

Modules		*ProbAttention*		*ProbSparse*		*LogSparse*		Auto-Correlation		LSH-Attention		Self-Attention	
L_x	L_y	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	336	**0.302**	**0.324**	0.481	0.472	0.362	0.413	0.339	0.372	0.366	0.404	0.375	0.425
	720	**0.382**	**0.381**	0.822	0.559	0.539	0.522	0.422	0.419	0.502	0.475	0.537	0.502
	1440	**0.429**	**0.446**	0.715	0.586	0.582	0.529	0.555	0.496	0.663	0.567	0.667	0.589
192	336	**0.318**	**0.332**	0.404	0.425	0.420	0.450	0.355	0.392	0.407	0.421	0.450	0.470
	720	**0.386**	**0.392**	1.148	0.654	0.552	0.513	0.429	0.430	0.636	0.571	0.554	0.533
	1440	**0.427**	**0.428**	0.732	0.602	0.958	0.736	0.503	0.484	1.069	0.756	–	–
336	336	**0.398**	**0.356**	0.417	0.434	0.474	0.474	0.361	0.406	0.442	0.476	0.501	0.485
	720	**0.412**	**0.393**	0.631	0.528	0.601	0.524	0.425	0.440	0.615	0.532	0.647	0.491
	1440	**0.494**	**0.445**	1.133	0.691	–	–	0.574	0.534	–	–	–	–

of univariate forecasting on full dataset of ETT in Table 2. Here ETT data covers two subsets with the only difference of observed interval [Zhou *et al.*, 2021]: ETTh_1/ETTh_2 are recorded hourly, and ETTm_1/ETTm_2 are recorded every fifteen minutes. Since FEDformer-f and FEDformer-w are presented as the best existing baseline models [Zhou *et al.*, 2022], we here primarily compare with them. As can be seen, Table 2 highlights the best results in bold, indicating AVGNets achieve best performance on all ETT dataset. To average all the results with different forecasting horizon, the improvement of AVGNets is by 21.4% on MSE and 19.8% on MAE compared with FEDformer.

5 Ablation Study

Since the angular weighting of visibility networks and *ProbAttention* module are two core components in the structure of AVGNets, we conduct respective ablation studies to explore the performance of each module. The operation is to remove each component at a time in our model, and AVGNets without different modules are named as

- **AVGNets/**w indicates AVGNets remove weighted angular operation and adopt vanilla visibility graph.
- **AVGNets/**p means AVGNets remove *ProbAttention* and adopt original dot-product self-attention block.
- **AVGNets/**wp refers to AVGNets remove both weight and *ProbAttention*, which is implemented by incorporating vanilla visibility graph into standard transformer architecture to eliminate the effect of angular weighting and *ProbAttention* module.

Results are together summarized in Table 3, and detailed comparisons are given below.

5.1 Performance of Angular Weighted Graph

Firstly, we compare AVGNets/wp with attention-based models. As shown in Table 3, when compared with Informer considered as vanilla transformer model, AVGNets/wp greatly improve the accuracy. Besides, AVGNets/wp also outperform FEDformer, the best baseline of attention-based model, indicating the introduction of visibility structure into attention mechanism contributes to forecasting time series. Furthermore, it can be seen AVGNets/p framework is superior to AVGNets/wp, which suggests angular weighted process further enhance predictive power than only considering binary visibility graph.

5.2 Performance of *ProbAttention* Module

To study the effect of *ProbAttention* module, likewise, we first look at the performance of AVGNets/p and AVGNets. For instance, in terms of MSE, AVGNets improve the accuracy by 15.1%, 24.0%, 22.8%, 21.5%, 28.3%, 25.3%, which indicates *ProbAttention* module performs better than self-attention mechanism in weighted graph networks. Besides, comparison of AVGNets/w and AVGNets/wp indicates that *ProbAttention* module also takes effect on classical visibility graph networks, greatly improving the performance.

Furthermore, we conduct experiments to compare *ProbAttention* module with other variant self-attention mechanisms, including $ProbSparse$ [Zhou *et al.*, 2021], $LogSparse$ [Li *et al.*, 2019], Auto-Correlation [Wu *et al.*, 2021] and Locality-Sensitive Hashing (LSH) Attentions, by replacing *ProbAttention* block in AVGNets with them. Results are listed in Table 4, from which we can conclude that *ProbAttention* is more efficient in predicting time series with longer length.

6 Conclusion

In this paper, different from existing classical statistical and modern deep learning models, we investigate time series from a novel perspective of visibility graph networks, and develop AVGNets to capture long-term dependency in variant graph structure. The novel *ProbAttention* is introduced to measure local and global similarity distribution of networks, consequently capturing inner and outer temporal correlation across multi-layer graphs. Compared with attention-based and graph-based methods, AVGNets have better capacity to handle non-stationary and non-linear patterns, together with higher stability. The proposed variant framework is superior to capturing long-term dependency, and also with higher interpretability in practical application. Hopefully our work can motivate further research on forecasting tasks from network perspective. Future work will consider directional edges for mining structural information within topological networks.

References

[Bollerslev, 1986] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[Box *et al.*, 2015] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[Chauhan *et al.*, 2022] Jatin Chauhan, Aravindan Raghuveer, Rishi Saket, Jay Nandy, and Balaraman Ravindran. Multi-variate time series forecasting on variable subsets. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 76–86, 2022.

[Chen *et al.*, 2021a] Yuzhou Chen, Ignacio Segovia, and Yulia R Gel. Z-gcnets: time zigzags at graph convolutional networks for time series forecasting. In *International Conference on Machine Learning*, pages 1684–1694. PMLR, 2021.

[Chen *et al.*, 2021b] Yuzhou Chen, Ignacio Segovia-Dominguez, Baris Coskunuzer, and Yulia Gel. Tamps2gcnets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *International Conference on Learning Representations*, 2021.

[De Gooijer and Hyndman, 2006] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Iacovacci and Lacasa, 2016] Jacopo Iacovacci and Lucas Lacasa. Sequential visibility-graph motifs. *Physical Review E*, 93(4):042309, 2016.

[Iacovacci and Lacasa, 2019] Jacopo Iacovacci and Lucas Lacasa. Visibility graphs for image processing. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):974–987, 2019.

[Jin *et al.*, 2022] Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pages 10280–10297. PMLR, 2022.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kitaev *et al.*, 2020] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[Lacasa and Iacovacci, 2017] Lucas Lacasa and Jacopo Iacovacci. Visibility graphs of random scalar fields and spatial data. *Physical Review E*, 96(1):012318, 2017.

[Lacasa *et al.*, 2008] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuño. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences*, 105(13):4972–4975, 2008.

[Lacasa *et al.*, 2015] Lucas Lacasa, Vincenzo Nicosia, and Vito Latora. Network structure of multivariate time series. *Scientific reports*, 5(1):1–9, 2015.

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

[Lim and Zohren, 2021] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[Lin *et al.*, 2021] Yang Lin, Irena Koprinska, and Mashud Rana. Ssdnet: State space decomposition neural network for time series forecasting. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 370–378. IEEE, 2021.

[Liu and Lü, 2010] Weiping Liu and Linyuan Lü. Link prediction based on local random walk. *EPL (europhysics Letters)*, 89(5):58007, 2010.

[Liu *et al.*, 2021] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2021.

[Mao and Zeng, 2023] Shengzhong Mao and Xiao-Jun Zeng. Simvgnets: Similarity-based visibility graph networks for carbon price forecasting. *Expert Systems with Applications*, page 120647, 2023.

[Mao *et al.*, 2021] Shengzhong Mao, Ching-Hsun Tseng, Jiayu Shang, Yuping Wu, and Xiao-Jun Zeng. Construction cost index prediction: A visibility graph network method. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2021.

[Olamat *et al.*, 2022] Ali Olamat, Pinar Ozel, and Aydin Akan. Synchronization analysis in epileptic eeg signals via state transfer networks based on visibility graph technique. *International Journal of Neural Systems*, 32(02):2150041, 2022.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

[Sims, 1980] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.

[Stankeviciute *et al.*, 2021] Kamile Stankeviciute, Ahmed M Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. *Advances in Neural Information Processing Systems*, 34:6216–6228, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[Xuan *et al.*, 2022] Qi Xuan, Jinchao Zhou, Kunfeng Qiu, Zhuangzhi Chen, Dongwei Xu, Shilian Zheng, and Xiaoniu Yang. Avgnet: Adaptive visibility graph neural network and its application in modulation classification. *IEEE Transactions on Network Science and Engineering*, 9(3):1516–1526, 2022.

[Zeng and Tang, 2022] Jie Zeng and Jinjun Tang. Modeling dynamic traffic flow as visibility graphs: A network-scale prediction framework for lane-level traffic flow based on lpr data. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[Zeng *et al.*, 2021] Sebastian Zeng, Florian Graf, Christoph Hofer, and Roland Kwitt. Topological attention for time series forecasting. *Advances in Neural Information Processing Systems*, 34:24871–24882, 2021.

[Zhao *et al.*, 2020] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? In *International Conference on Machine Learning*, pages 11365–11375. PMLR, 2020.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*, 2022.

[Zou *et al.*, 2019] Yong Zou, Reik V Donner, Norbert Marwan, Jonathan F Donges, and Jürgen Kurths. Complex network approaches to nonlinear time series analysis. *Physics Reports*, 787:1–97, 2019.