FV-MgNet: Fully Connected V-cycle MgNet for Interpretable Time Series Forecasting

Anonymous authors
Paper under double-blind review

Abstract

By investigating iterative methods for a constrained linear model, we propose a new class of fully connected V-cycle MgNet for long-term time series forecasting, which is one of the most difficult tasks in forecasting. MgNet is a CNN model that was proposed for image classification based on the multigrid (MG) methods for solving discretized partial differential equations (PDEs). We replace the convolutional operations with fully connected operations in the existing MgNet and then apply them to forecasting problems. Motivated by the V-cycle structure in MG, we further propose the FV-MgNet, a V-cycle version of the fully connected MgNet, to extract features hierarchically. By evaluating the performance of FV-MgNet on popular data sets and comparing it with state-of-the-art models, we show that the FV-MgNet achieves better results with less memory usage and faster inference speed. In addition, we also develop ablation experiments to demonstrate that the structure of FV-MgNet is the best choice among the many variants.

1 Introduction

Long-term time series forecasting has played an important role in numerous applications across an array of sectors, including retail (Böse et al. (2017); Courty & Li (1999)), healthcare (Lim et al. (2018); Zhang & Nawata (2018)), and engineering (Zhang et al. (2019); Gonzalez-Vidal et al. (2019)). Various deep learning models have been developed for time series forecasting, among which recurrent neural networks (RNN) are probably the most extensively studied (Connor et al. (1994); Hewamalage et al. (2021)). Recently, following its success in natural language processing (NLP) and computer vision (CV) research (Vaswani et al. (2017); Devlin et al. (2019); Dosovitskiy et al. (2021); Rao et al. (2021)), the Transformer model has become one of the most popular research directions in time series forecasting (Li et al. (2019); Zhou et al. (2021); Wu et al. (2021)). Efficient self-attention has been introduced, and many structures have been proposed to handle time-series forecasting tasks, such as decomposing the sequence (Wu et al. (2021)), designing a special linear attention structure (Li et al. (2019); Zhou et al. (2021)), and applying Fourier transform (Wu et al. (2021)) or wavelet transform (Zhou et al. (2022b)) to the self-attention structure. However, the Transformer model has a high computational cost, and many experiments have found that forecasting results degenerate as the length of the input sequence increases (Wen et al. (2022)).

By investigating iterative methods for a constrained linear model, we propose a new class of fully connected V-cycle MgNet for long-term time series forecasting. This is the first time an MgNet type model has successfully been applied to this field. One of the main technologies this model is used for is to replace the convolutional operations with fully connected operations in the existing MgNet (He & Xu (2019)), which is a CNN model proposed for image classification based on the multigrid (MG) methods (Xu (1992); Xu & Zikatanov (2002; 2017)) for solving discretized partial differential equations (PDEs). The other technology, that we propose, is a V-cycle version of the fully connected MgNet to extract features hierarchically, motivated by the V-cycle structure in MG, where different frequency information can be captured through the size transformation between grids. By evaluating the performance of the FV-MgNet on popular data sets and comparing it with state-of-the-art models, we present that the FV-MgNet achieves better results with much less memory usage and faster inference speed (as shown in Figure 1 and Table 1). We also demonstrate that numerical results do not degenerate as the length of the input increases (as shown in Table 3).
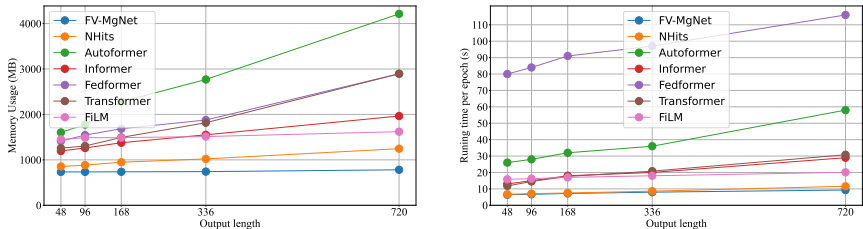
Figure 1: Computational efficiency benchmark on the Exchange multivariate dataset, batch size is 16. Time usage is measured for one epoch training time. Left: Memory usage as output length, input length is fixed to 96. Right: Epoch time usage as output length, input length is fixed to 96.

Table 1: Parameters for the Exchange dataset with input length and output length fixed to 96.

Methods	FV-MgNet	FiLM	N-HiTS	FEDformer	Autoformer	Informer	Transformer
Params (M)	0.12	9.44	1.73	16.31	12.14	12.45	11.66

This paper is organized as follows. In section 2, we review some of the related work. In section 3, we first consider the time series forecasting problem as a constrained linear model and solve it using the multigrid method, an iterative method, and then we introduce FV-MgNet models for forecasting problems. Numerical experiments and a Conclusion are given in Sections 4 and 5, respectively.

2 RELATED WORK

Recurrent neural networks are widely used in time series prediction problems. LSTM (Graves (2012)) and GRU (Chung et al. (2014)) can alleviate the inherent problems such as gradient disappearance and gradient explosion through a gating mechanism. Many works based on LSTM and GRU have proposed numerous time series prediction models (e.g. (Wen et al. (2017); Salinas et al. (2020); Guo et al. (2019); Fan et al. (2019)). However, the single-step prediction method cannot accurately capture long-term time series features, and it is difficult to parallelize. More recently, a convolutional neural network has been applied to this task and a temporal convolutional network (TCN) was proposed (Sen et al. (2019)). Although this method is efficient, there is also a prediction bottleneck in the long-term time series prediction problem. With the innovation of Transformers in NLP and CV (Vaswani et al. (2017); Devlin et al. (2019); Dosovitskiy et al. (2021)), Transformer-based models have also been proposed, and perform well, in time series forecasting (Li et al. (2019); Zhou et al. (2021); Wu et al. (2021)). Transformer-based models excel in modeling long-term dependencies for sequential data. Furthermore, many efficient Transformers are designed in time series forecasting to overcome the quadratic computation complexity of the original Transformer without performance degradation. LogTrans (Li et al. (2019)) adopts log-sparse attention and achieves $N \log^2 N$ complexity. Reformer (Kitaev et al. (2020)) introduces a local-sensitive hashing that reduces the complexity to $N \log N$. Informer (Zhou et al. (2021)) uses a KL-divergence-based method to select top-k in the attention matrix and costs $N \log N$ in complexity. Most recently, Autoformer (Wu et al. (2021)) introduced an auto-correlation block in place of canonical attention to perform sub-series level attention, which achieves $N \log N$ complexity with the help of Fast Fourier transform (FFT) and top-k selection in the auto-correlation matrix. Fedformer (Zhou et al. (2022b)) designs frequency domain attention by applying Fourier or wavelet transform. ETSformer (Woo et al. (2022)) proposes exponential smoothing attention (ESA) and frequency attention (FA), which improves accuracy and efficiency.

Recently, many non-Transformer models have also achieved good results in forecasting problems. FiLM (Zhou et al. (2022a)) applies Legendre Polynomial projections, Fourier projections, and low-rank approximations in time series forecasting. The NBEATS (Oreshkin et al. (2019)) model was proposed to design a double residual structure with fully connected layers, which has achieved good results in univariate prediction. N-HiTS (Challu et al. (2022)) adds a multi-scale structure to NBEATs, so that it can better capture the information from different periods in the sequence and achieve improved results in a multivariate model. In NBEATS and N-HiTS, we observe fully connected

layers; multi-scale structures are important to help improve model performance. MgNet (He & Xu (2019)) is a CNN model proposed for image classification based on the multigrid, which has been a successful method for solving PDEs. MgNet is explained by using multigrid ideas to solve a constrained linear model (He et al. (2021)) so that it has a well-designed multiscale structure. We investigate the fully connected layers and multiscale structure under the MgNet framework to solve forecasting problems.

3 METHODOLOGY

In this section, we present our proposed model, FV-MgNet, for time series forecasting using the constrained linear model and a multigrid structure; a detailed description of FV-MgNet is given in Algorithm 1. Considering the data-label pair as (f, y) where $f \in \mathbb{R}^I$ is the input time series and $y \in \mathbb{R}^O$ is the predicted time series, we propose the following feature extraction and interpolation scheme as an interpretable model to build the mapping from f to y:

$$\text{Input: } f \xrightarrow[\text{constrained linear model: } Au=f]{\text{feature extraction}} \text{Feature: } u \xrightarrow[\text{neural network approximation}]{\text{feature interpolation: } \mathcal{I}} \text{Output: } y. \qquad (1)$$

To summarise, we have

$$y = \mathcal{I}(u(f)). \qquad (2)$$

Here, we denote $u \in \mathbb{R}^{I'}$ as the feature vector. In general, the information contained in the feature (essential dimension) should be less than or equal to the information from the data (embedded dimension), such as in, for example, manifold learning (Tenenbaum et al. (2000); Roweis & Saul (2000); Donoho & Grimes (2003)); precisely, $I' \leq I$. To extract more intrinsic features but retain as much information as possible, we take $I' = I$ while adopting a V-cycle structure (similar to U-Net architecture) for feature extraction. First, we propose to apply the constrained liner model (He et al. (2021)), which was originally designed for vision tasks for feature extraction. Then, we propose to apply the fully connected operator as the underlying linear model instead of the convolutional operator in image processing. Finally, we introduce the multigrid structure to construct features in multi-resolution and reduce the computational complexity. After obtaining the feature vector u, we propose to approximate $\mathcal{I}(u)$ by using a one-hidden layer ReLU neural network that can be expressed as

$$\mathcal{I}(u) \approx W^2 \sigma \left(W^1 u \right). \qquad (3)$$

The proposed model uses the multigrid method as the main framework, performs residual correction on the input of different resolutions, integrates the information on other grids through the extension operator, and then adopts a one-hidden layer ReLU neural network approximation for interpolation to obtain the long horizon accurate predictions. In addition, compared to models based on transformer architecture, our model is implemented with fully connected layers, which has a more elegant structure with higher computational efficiency and a lower memory overhead.

3.1 CONSTRAINED LINEAR MODEL: AN INTERPRETABLE MODEL FOR FEATURE EXTRACTION

The constrained linear model (He et al. (2021)) was originally proposed for vision tasks; in this work, we demonstrate that it can also be applied as an interpretable model for feature extraction in time series problems. For any input time series $f \in \mathbb{R}^I$ in data space, the constrained linear model is defined as the following data-feature mapping

$$Au = f, \qquad (4)$$

where $u \in \mathbb{R}^{I'}$ is the underlying feature vector such that

$$[u]_i \geq 0, \quad i = 1 : I'. \qquad (5)$$

Here, we state that $A : \mathbb{R}^I \mapsto \mathbb{R}^{I'}$ is a linear (affine) operator from the feature space to the data space. It is crucial to solving the linear equation 4 with a nonlinear constraint equation 5. Typically, the iterative method is an important part of solving algebraic systems. For example, given an algebraic system $Au = f$, the solution can be obtained through the residual correction method

$$u^{i+1} = u^i + B^i(f^i - Au^i), \quad i = 1 : \nu, \qquad (6)$$

where $B^i : \mathbb{R}^{I'} \mapsto \mathbb{R}^I$ is an approximation of the inverse of the data-feature mapping A. Recalling the definition of the activation function $\sigma(x) = \mathrm{ReLU}(x) := \max\{0, x\}$, the above iterative process can be naturally modified to preserve the constraint equation 5:

$$u^i = u^{i-1} + \sigma \circ B^i \sigma \circ (f - Au^{i-1}), \quad i = 1 : \nu. \tag{7}$$

Here, $\sigma \circ (\cdot)$ denotes the element-wise applications of the nonlinear activation function σ. As demonstrated in (He et al. (2021)), the above iteration scheme can degenerate to the basic block of ResNet (He et al. (2016)) if $r = f - Au$ is chosen as the feature. However, theoretical arguments and numerical results in (He et al. (2021)) indicate that u should be a more appropriate choice as the feature. In this work, we also take equation 7 as the basic block in our model for feature extraction.

3.2 Underlying system A: dense or sparse?

In vision tasks, A and B^i are generally chosen as learnable multi-channel convolution operators. For example, the basic block for image classification in MgNet (He & Xu (2019)) is:

$$u^i = u^{i-1} + \sigma \circ B^i * \sigma \circ \left(f - A * u^{i-1} \right),$$

where A and B^i are both convolution operators. Generally, convolutional operators are also used in time series processing, from the frequency perspective, for signal decomposition, especially in classical methods such as wavelet-based methods (Percival & Walden (2000); Joo & Kim (2015)). However, in long-term time series forecasting problems, such as the data sets that we will use in §A.1, the time series can be highly complex. For example, Figure 2 shows three typical examples in three different data sets, which can vary significantly in frequencies and amplitudes. Thus, convolutional
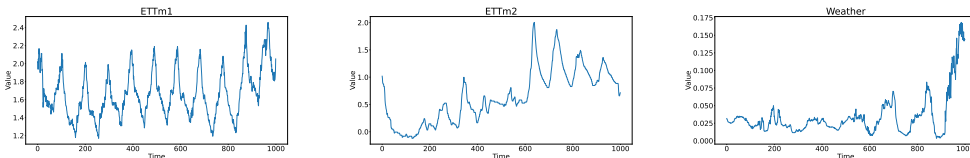

Figure 2: Some typical long-term time series from different data sets.

operators cannot process these different data sets uniformly. A more appropriate choice for the underlying system A is to consider it as a general linear mapping without any special structures. Consequently, we use a fully connected layer to learn the system A. Accordingly, we also choose B^i as a fully connected layer. Thus, we have

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} \circ \left(f^\ell - A^\ell u^{\ell,i-1} \right), \quad B^{\ell,i} \in \mathbb{R}^{I_\ell \times I_\ell}, A^\ell \in \mathbb{R}^{I_\ell \times I_\ell}. \tag{8}$$

For simplicity, we use $A^\ell \in \mathbb{R}^{I_\ell \times I_\ell}$ and $B^{\ell,i} \in \mathbb{R}^{I_\ell \times I_\ell}$ to denote the fully connected operators with bias, respectively. Numerical results in Table 5 demonstrate that the best combination of A and B^i is the fully connected operator.

3.3 Multigrid structure: a hierarchical feature extraction scheme

A hierarchical feature extraction scheme for time series processing is typically standard in classical and CNN-based methods for imaging processing. For time series signals, the multilevel structure (multi-resolution) can help to capture different features of the signal with different frequencies. For example, N-HiTS (Challu et al. (2022)) introduced the hierarchical structure into the forecast module in N-BEATS (Oreshkin et al. (2019)), achieving a higher accuracy with fewer parameters.

Here, we consider the multigrid structure, which is a more mature and relevant multi-level method to solve linear systems arising from numerical PDEs (Xu (1992); Xu & Zikatanov (2002; 2017)). In the multigrid method, we first have a series of grids (resolutions) for the feature and data space. More precisely, let us denote $I_\ell = \frac{I}{2^{\ell-1}}$ as the grid size (resolution) for the ℓ-th level grid. Correspondingly, we have constrained linear models on different grids,

$$A^\ell u^\ell = f^\ell, \tag{9}$$

4

where u^ℓ, f^ℓ, and A^ℓ represent the feature, data, and underlying system in the ℓ-th level respectively. By applying the numerical scheme in equation 7, we have the basic block in the ℓ-th level

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} \sigma(f^\ell - A^\ell u^{\ell,i-1}), \quad i = 1 : \nu_\ell. \tag{10}$$

Here, ν_ℓ is the number of iterations on each level, which is a hyper-parameter of the model. Notably, there is only one A^ℓ on the ℓ-th level, while we have ν_ℓ different $B^{\ell,i}$ to learn. This is a fundamental assumption of the constrained linear model; specifically, that there can only be one system operator A^ℓ on each level, while there can be different solving operators $B^{\ell,i}$ for solving the model $A^\ell u^\ell = f^\ell$.

To integrate different grids, restriction and prolongation operators must be involved, such as different pooling operations in typical CNN models. Recently, MgNet (He & Xu (2019)) has successfully applied the multigrid structure to image classification problems, using convolution with $stride = 2$ as the restriction operator to achieve a multi-scale transformation. The scale transformation for feature u and data f is defined as follows:

$$\begin{aligned}
u^{\ell+1,0} &= \Pi_\ell^{\ell+1} *_2 u^{\ell,\nu_\ell} \\
f^{\ell+1} &= R_\ell^{\ell+1} *_2 (f^\ell - A^\ell * u^{\ell,\nu_\ell}) + A^{\ell+1} * u^{\ell+1,0},
\end{aligned} \tag{11}$$

where $\Pi_\ell^{\ell+1} *_2$ and $R_\ell^{\ell+1} *_2$ are convolution operators with $stride = 2$. For the prolongation process in image classification, MgNet uses the deconvolution operator $\Pi_{\ell+1}^\ell *^2$ with $stride = 2$, which interpolates information on a coarse grid to a fine grid.

In our model, considering the fully connected structure, we take the restriction (sub-sampling) and prolongation (interpolation) as linear mappings between $\mathbb{R}^{I_\ell}$ and $\mathbb{R}^{I_{\ell+1}}$. More precisely, we take $\Pi_\ell^{\ell+1}, R_\ell^{\ell+1} \in \mathbb{R}^{I_{\ell+1} \times I_\ell}$ and $\Pi_{\ell+1}^\ell \in \mathbb{R}^{I_\ell \times I_{\ell+1}}$ as fully connected layers (with bias) for restoration and prolongation operators.

Another practical reason to introduce the multigrid structure is to further reduce the computational complexity of the proposed model. Numerical results in Table 4 show the efficiency and improvements obtained by introducing the multigrid structure.

3.4 FV-MgNet: A fully connected V-cycle MgNet model for time series forecasting

By combining the feature extraction process discussed above and the feature interpolation approximation in equation 3, we have the FV-MgNet algorithm for time series forecasting.

Here, $u = u^{1,\nu_1}$ is the feature extracted by the fully-connected constrained linear model with the multigrid scheme. In addition, we take $W_1 \in \mathbb{R}^{I \times O}$ and $W_2 \in \mathbb{R}^{O \times O}$ as the fully connected layers (with bias) to construct the one-hidden layer approximation for the feature interpolation $\mathcal{I}(u)$.

4 EXPERIMENTS

To verify the performance of the FV-MgNet, we conducted experiments on six popular real-world data sets, spanning the domains electricity, economy, traffic, weather, and disease. The models we compared included Transformer models, such as FEDformer (Zhou et al. (2022b)), ETSformer (Woo et al. (2022)), Autofomer (Wu et al. (2021)), the fully connected layer-based model N-HiTS (Challu et al. (2022)), and the frequency enhanced model FiLM (Zhou et al. (2022a)).

4.1 EXPERIMENT RESULTS

Forecasting result As shown in Table 2, FV-MgNet achieves the best performance on the six data sets when compared with the other models, except for FiLM on Exchange and Traffic. Compared with ETSformer, which achieves state-of-the-art results in Transformer type models, FV-MgNet reduces MSE error by more than 18%; improved results were particularly obvious on the traffic and ILI data sets, in which MSE decreases of 31% and 29% MSE, respectively, were observed. Compared with N-HiTS, which uses the fully connected layer as the basic block, FV-MgNet reduces MSE by 8%. Although FV-MgNet and FiLM have a comparable performance on Exchange and Traffic, FV-MgNet outperforms FiLM by 5% MSE and 6% MAE on the other data sets. Notably, FV-MgNet achieves

---

Algorithm 1 $y = \text{FV-MgNet}(f, J, \nu_\ell, I, O)$

1: **Input**: series $f \in \mathbb{R}^I$, number of grids J, number of smoothing iterations ν_ℓ for $\ell = 1 : J$, input length I, output length O.
2: Initialization: $f^1 = Wf$, $W \in \mathbb{R}^{I \times I}$, $u^{1,0} = 0$, $I_\ell = \frac{I}{2^{\ell-1}}$
3: **for** $\ell = 1 : J$ **do**
4:     **for** $i = 1 : \nu_\ell$ **do**
5:         Feature extraction (smoothing):

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} \circ \left(f^\ell - A^\ell u^{\ell,i-1} \right), \quad B^{\ell,i} \in \mathbb{R}^{I_\ell \times I_\ell}, A^\ell \in \mathbb{R}^{I_\ell \times I_\ell}$$

6:     **end for**
7:     Note: $u^\ell = u^{\ell,\nu_\ell}$
8:     Restriction:

$$u^{\ell+1,0} = \Pi_\ell^{\ell+1} u^\ell, \quad \Pi_\ell^{\ell+1} \in \mathbb{R}^{I_{\ell+1} \times I_\ell}$$

$$f^{\ell+1} = R_\ell^{\ell+1}(f^\ell - A^\ell u^\ell) + A^{\ell+1} u^{\ell+1,0}, \quad R_\ell^{\ell+1} \in \mathbb{R}^{I_{\ell+1} \times I_\ell}$$

9: **end for**
10: **for** $\ell = J - 1 : 1$ **do**
11:     Prolongation

$$u^{\ell,0} = u^\ell + \Pi_{\ell+1}^\ell(u^{\ell+1} - u^{\ell+1,0}), \quad \Pi_{\ell+1}^\ell \in \mathbb{R}^{I_\ell \times I_{\ell+1}}$$

12:     **for** $i = 1 : \nu_\ell$ **do**
13:         Feature extraction (smoothing):

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ \overline{B}^{\ell,i} \circ \left(f^\ell - \overline{A}^\ell u^{\ell,i-1} \right), \quad \overline{B}^{\ell,i} \in \mathbb{R}^{I_\ell \times I_\ell}, \overline{A}^\ell \in \mathbb{R}^{I_\ell \times I_\ell}$$

14:     **end for**
15: **end for**
    Feature Interpolation:

$$y = W^2 \sigma \left(W^1 u^{1,\nu_1} \right), \quad u^{1,\nu_1} \in \mathbb{R}^I, \ W_1 \in \mathbb{R}^{I \times O}, \ W_2 \in \mathbb{R}^{O \times O}$$

---

the best results in the longest time-series prediction of all datasets (ILI with an output length 60 and the other data sets with an output length of 720), which indicates our model is more suitable for long-range prediction compared to other models.

Prediction results visualization In Figure 3, we plot the prediction results of four models (FV-MgNet, N-HiTS, FEDformer, and Autoformer) on three data sets with different temporal patterns: ETTh1 (testset sequence 650, variable 7), ETTh2 (testset sequence 150, variable 7), and Weather (testset sequence 185, variable 20). On ETTh1, FV-MgNet can accurately predict seasonal and trend information and N-HiTS can partially capture trend information, whereas FedFormer and Autoformer can only capture seasonal information. In ETTh2, all four methods can accurately capture season information, and FV-MgNet can capture trend information more accurately than the others. In Weather, FV-MgNet can predict both season and trend information accurately, N-HiTS can partially capture trend information, whereas FEDformer and Autofomer cannot predict either season or trend information.

Computational efficiency We measured the number of parameters of FV-MgNet and the other models on the multivariate Exchange data set. As shown in Table 1, when the input length and output length are both set to 96, the total number of parameters of FV-MgNet is only 1% of those of the Transformer type model and FiLM, and 10% of N-HiTS. We also recorded memory usage and running time of each epoch during the training process in Figure 1, where we set the input length to 96 and changed different output lengths. Both memory usage and running time of FV-MgNet are the smallest. The running time of FV-MgNet is ten times faster than Autoformer and two times faster than FiLM. The running time of FV-MgNet is ten times faster than Autoformer, two times faster than FiLM, and 10% faster than N-HiTS. The memory usage of FV-MgNet is five times less than Autoformer, two times less than FiLM, and 1.5 times less than N-HiTS. Furthermore, the memory usage and running time of FV-MgNet are almost quasi-constant as the output length increases.

Table 2: Multivariate results with different prediction lengths $O \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. The best results are highlighted in bold.

| Methods		FV-MgNet		FiLM		N-HiTS		ETSformer		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	0.173	**0.253**	**0.165**	0.256	0.176	0.255	0.189	0.280	0.203	0.287	0.255	0.339
	192	0.230	**0.296**	**0.222**	**0.296**	0.245	0.305	0.253	0.319	0.269	0.328	0.281	0.340
	336	0.279	**0.329**	**0.277**	0.333	0.295	0.346	0.314	0.357	0.325	0.366	0.339	0.372
	720	**0.367**	**0.385**	0.371	0.389	0.401	0.426	0.414	0.413	0.421	0.415	0.422	0.419
Electricity	96	**0.144**	**0.250**	0.154	0.267	0.147	0.249	0.187	0.304	0.183	0.297	0.201	0.317
	192	**0.163**	0.262	0.164	**0.258**	0.167	0.269	0.199	0.315	0.195	0.308	0.222	0.334
	336	**0.176**	**0.276**	0.188	0.283	0.186	0.290	0.212	0.329	0.212	0.313	0.231	0.338
	720	**0.212**	**0.308**	0.249	0.338	0.243	0.340	0.233	0.345	0.231	0.343	0.254	0.361
Exchange	96	0.082	0.206	**0.079**	**0.204**	0.092	0.211	0.085	0.204	0.139	0.276	0.197	0.323
	192	0.184	0.314	**0.159**	**0.292**	0.208	0.322	0.182	0.303	0.256	0.369	0.300	0.369
	336	0.307	0.416	**0.270**	**0.398**	0.371	0.443	0.428	0.428	0.426	0.464	0.509	0.524
	720	**0.554**	**0.582**	0.830	0.721	0.888	0.723	1.025	0.774	1.090	0.800	1.447	0.941
Traffic	96	**0.396**	**0.285**	0.416	0.294	0.402	0.282	0.607	0.392	0.562	0.349	0.613	0.388
	192	0.417	0.295	**0.408**	**0.288**	0.420	0.297	0.621	0.399	0.562	0.346	0.616	0.382
	336	0.436	0.302	**0.425**	**0.298**	0.448	0.313	0.622	0.396	0.570	0.323	0.622	0.337
	720	**0.468**	**0.315**	0.520	0.353	0.539	0.353	0.632	0.396	0.596	0.368	0.660	0.408
Weather	96	**0.155**	**0.196**	0.199	0.262	0.158	0.195	0.197	0.281	0.217	0.296	0.266	0.336
	192	**0.201**	**0.239**	0.228	0.288	0.211	0.247	0.237	0.312	0.276	0.336	0.307	0.367
	336	**0.244**	**0.279**	0.267	0.323	0.274	0.300	0.298	0.353	0.339	0.380	0.359	0.395
	720	**0.313**	**0.329**	0.319	0.361	0.351	0.353	0.352	0.388	0.403	0.428	0.419	0.428
ILI	24	**1.647**	**0.764**	1.970	0.875	1.862	0.869	2.527	1.020	2.203	0.963	3.483	1.287
	36	**1.841**	**0.839**	1.982	0.859	2.071	0.969	2.615	1.007	2.272	0.976	3.103	1.148
	48	**1.831**	**0.853**	1.868	0.896	2.346	1.042	2.359	0.972	2.209	0.981	2.669	1.085
	60	**1.765**	**0.814**	2.057	0.929	2.560	1.073	2.487	1.016	2.545	1.061	2.770	1.125
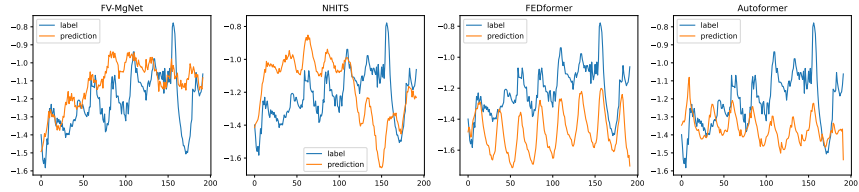
(a) $ETTh1$
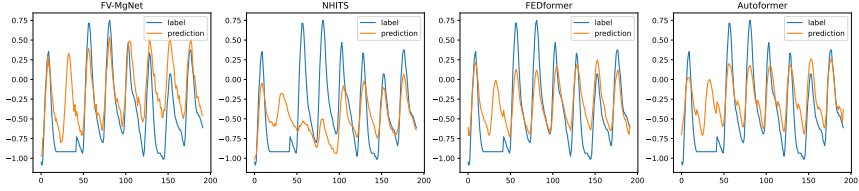
(b) $ETTh2$
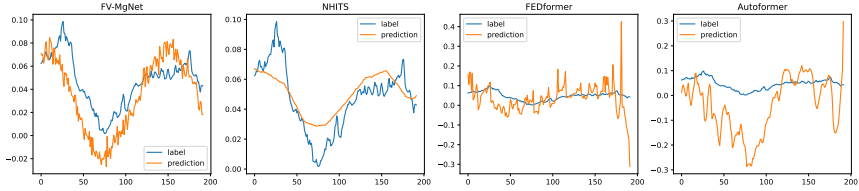
(c) $Weather$

Figure 3: Visualization of the forecasting output (Y-axis) of four models (FV-MgNet, N-HiTS, FEDformer, and Autoformer) with an output length $O = 192$ (X-axis) on the ETTh1, ETTh2,and Weather data sets, respectively.

7

Robustness Intuitively, if the output length is fixed, better results should be otained from a longer input length. We set the input length to 96 and increased it from 96 to 720. As shown in Table 3, the prediction results are reduced when we increase the input length of the Transformer type models. However, FV-MgNet maintains stable prediction results on the ETT data set and even obtains a significant improvement on the Weather data set. The results show that FV-MgNet can effectively capture the features in long input sequences, and the robustness of FV-MgNet is significantly better than Transformer models in long input sequences.

Table 3: The MSE comparisons in robustness experiment of forecasting 96 steps for the $ETTm2$ and weather data sets with prolonged input length.

Methods		FV-MgNet	FEDformer	Autoformer	Informer	Reformer	LogFormer	Transformer
Metric		MSE	MSE	MSE	MSE	MSE	MSE	MSE
ETTm2	96	0.176	0.203	0.239	0.428	0.615	0.667	0.557
	192	0.173	0.275	0.265	0.385	0.686	0.697	0.710
	336	0.173	0.347	0.375	1.078	1.359	0.937	1.078
	720	0.180	0.392	0.315	1.057	1.443	2.153	1.691
Weather	96	0.185	0.217	0.266	0.300	0.689	0.458	0.604
	192	0.167	0.253	0.276	0.325	0.724	0.537	0.674
	336	0.155	0.278	0.336	0.346	0.736	0.742	0.692
	720	0.153	0.276	0.547	0.421	0.701	0.884	0.740

4.2 Ablation results

In this subsection, we investigated the structure of MgNet for forecasting problems. We first tested two different operators, convolutional operators or fully connected operators, for A and B under V-cycle structure MgNet. The results in Table 5 show that both A and B, using the fully connected operators, are significantly better than the other choices. This verifies that fully connected layers are more suitable for time series forecasting tasks than convolution layers under the MgNet framework. Next, we explored the structure of fully connected MgNet. We designed three models: (1) Residual network, which uses the same dimension for A and B in different grids (the algorithm is shown in Appendix Algorithm 2); (2) \-MgNet, which has only a single hierarchical structure (the algorithm is shown in Algorithm 3); and (3) FV-MgNet as shown in Algorithm 1. (3) FV-MgNet as shown in Algorithm 1. The results obtained using the residual network are poor but can be improved as a \-MgNet after adding a one-side hierarchical structure. However, numerical results demonstrate that FV-MgNet, MgNet with V-cycle (two-side hierarchical structure), is the best choice.

4.3 Other Studies

Details of other experimental studies are provided in the Appendix: (1) univariate forecasting experiments, (2) Transformer models experiments under the same setting, (3) parameter sensitivity experiments of grid number and iteration number, (4) experiments under different input lengths, and (5) ETT full benchmark experiments.

5 Conclusions

The iterative method and multigrid method in numerical PDEs have well-developed mathematical theories. By considering the forecasting problems from the constrained model point of view, and taking advantage of the multigrid method and MgNet, we propose FV-MgNet models for long-term time series forecasting. We investigated variants of FV-MgNet and found that the use of fully connected MgNet with V-cycle gives the best results. In comparison with state-of-the-art models, we demonstrate the good performance of FV-MgNet on different data sets, which achieves better results with fewer parameters and faster inference speed. Furthermore, the results obtained by FV-MgNet significantly improve as the length of the input sequence increases.

Table 4: Ablation experiments to compare FV-MgNet, \-MgNet, and Residual model. Our numerical results demonstrate that multigrid structure for feature extraction can indeed improve the accuracy of the method.

Methods		FV-MgNet		\-MgNet		Residual	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	**0.173**	**0.253**	0.183	0.270	0.212	0.318
	192	**0.230**	**0.296**	0.247	0.311	0.275	0.340
	336	**0.279**	**0.329**	0.321	0.356	0.354	0.385
	720	**0.367**	**0.385**	0.434	0.422	0.482	0.462
Electricity	96	**0.144**	**0.250**	0.179	0.287	0.213	0.312
	192	**0.163**	**0.262**	0.204	0.305	0.262	0.349
	336	**0.176**	**0.276**	0.216	0.317	0.272	0.370
	720	**0.212**	**0.308**	0.280	0.369	0.341	0.436
Exchange	96	**0.082**	**0.206**	0.093	0.223	0.104	0.283
	192	**0.184**	**0.314**	0.242	0.361	0.298	0.394
	336	**0.307**	**0.416**	0.358	0.455	0.443	0.552
	720	**0.554**	**0.582**	1.258	0.834	1.430	0.968
Traffic	96	**0.396**	**0.285**	0.717	0.398	0.589	0.348
	192	**0.417**	**0.295**	0.686	0.401	0.567	0.345
	336	**0.436**	**0.302**	0.661	0.388	0.553	0.342
	720	**0.468**	**0.315**	0.666	0.384	0.612	0.351
Weather	96	**0.155**	**0.196**	0.181	0.222	0.217	0.263
	192	**0.201**	**0.239**	0.218	0.259	0.255	0.308
	336	**0.244**	**0.279**	0.267	0.303	0.302	0.331
	720	**0.313**	**0.329**	0.335	0.345	0.389	0.403
ILI	24	**1.647**	**0.764**	3.123	1.130	3.261	1.280
	36	**1.841**	**0.839**	3.247	1.182	3.538	1.461
	48	**1.831**	**0.853**	2.890	1.063	3.295	1.198
	60	**1.765**	**0.814**	3.218	1.153	3.418	1.229

Table 5: Different encoding methods for A and B, marked as (A,B), the first element represents the A encoding method and the second element represents the B encoding method. The first row represents the selected encoding method, where Conv represents the convolution layer, and FC represents the fully connected layer.

(A,B)		(FC,FC)		(Conv,Conv)		(Conv,FC)		(FC,Conv)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	**0.173**	**0.253**	0.464	0.383	0.304	0.379	0.355	0.351
	192	**0.230**	**0.296**	0.608	0.610	0.438	0.540	0.448	0.366
	336	**0.279**	**0.329**	0.893	1.276	0.481	0.624	0.588	0.487
	720	**0.367**	**0.385**	1.151	2.350	0.824	1.319	0.737	0.628
Electricity	96	**0.144**	**0.250**	0.407	0.336	0.340	0.418	0.349	0.423
	192	**0.163**	**0.262**	0.410	0.336	0.335	0.420	0.447	0.498
	336	**0.176**	**0.276**	0.393	0.316	0.460	0.507	0.423	0.493
	720	**0.212**	**0.308**	0.409	0.335	0.384	0.457	0.379	0.447
Exchange	96	**0.082**	**0.206**	0.798	0.875	0.492	0.348	0.528	0.366
	192	**0.184**	**0.314**	0.977	1.364	0.615	0.447	0.428	0.474
	336	**0.307**	**0.416**	0.903	1.180	0.676	0.482	0.486	0.493
	720	**0.554**	**0.582**	1.873	2.469	1.506	1.092	1.629	1.086
Weather	96	**0.155**	**0.196**	0.256	0.174	0.174	0.258	0.168	0.251
	192	**0.201**	**0.239**	0.308	0.243	0.226	0.309	0.225	0.305
	336	**0.244**	**0.279**	0.351	0.289	0.321	0.362	0.296	0.365
	720	**0.313**	**0.329**	0.401	0.366	0.585	0.490	0.482	0.454

REFERENCES

Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12):1694–1705, 2017.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Jerome T Connor, R Douglas Martin, and Les E Atlas. Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2):240–254, 1994.

Pascal Courty and Hao Li. Timing of seasonal sales. *The Journal of Business*, 72(4):545–572, 1999.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, June 2-7, 2019*, pp. 4171–4186, 2019.

David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations (ICLR), Austria, May 3-7, 2021*, 2021.

Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, and Heng Huang. Multi-horizon time series forecasting with temporal attention learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2527–2535, 2019.

Aurora Gonzalez-Vidal, Fernando Jimenez, and Antonio F Gomez-Skarmeta. A methodology for energy multivariate time series forecasting in smart buildings based on feature selection. *Energy and Buildings*, 196:71–82, 2019.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring interpretable lstm neural networks over multi-variable data. In *International conference on machine learning*, pp. 2494–2504. PMLR, 2019.

Juncai He and Jinchao Xu. Mgnet: A unified framework of multigrid and convolutional neural network. *Science China Mathematics*, pp. 1–24, 2019.

Juncai He, Jinchao Xu, Lian Zhang, and Jianqing Zhu. An interpretive constrained linear model for resnet and mgnet. *arXiv preprint arXiv:2112.07441*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37 (1):388–427, 2021.

Tae Woo Joo and Seoung Bum Kim. Time series forecasting based on wavelet filtering. *Expert Systems with Applications*, 42(8):3868–3874, 2015.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. *NeurIPS*, 18:7483–7493, 2018.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Donald B Percival and Andrew T Walden. *Wavelet methods for time series analysis*, volume 4. Cambridge university press, 2000.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

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.

Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

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 (NeurIPS)*, 30, 2017.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 101–112, 2021.

Jinchao Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4): 581–613, 1992.

Jinchao Xu and Ludmil Zikatanov. The method of alternating projections and the method of subspace corrections in hilbert space. *Journal of the American Mathematical Society*, 15(3):573–597, 2002.

Jinchao Xu and Ludmil Zikatanov. Algebraic multigrid methods. *Acta Numerica*, 26:591–721, 2017.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1409–1416, 2019.

J Zhang and K Nawata. Multi-step prediction for influenza outbreak by an adjusted long short-term memory. *Epidemiology & Infection*, 146(7):809–816, 2018.

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 *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *arXiv preprint arXiv:2205.08897*, 2022a.

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*, 2022b.

A APPENDIX

A.1 DATASETS

We considered six benchmark data sets across diverse domains. The details of these data sets are summarized as follows. 1) The ETT[1] (Zhou et al. (2021)) data set contains two sub-datasets (ETT1 and ETT2) collected from two electricity transformers; there are two versions of each data set at different resolutions (15min and 1h), with multiple series of loads and one series of oil temperatures. 2) The Electricity[2] data set contains the electricity consumption of clients, with each column corresponding to one client. 3) The Exchange[3] (Lai et al. (2018)) data set contains the exchange of eight countries. 4) The Traffic[4] dataset contains the occupation rate of the freeway system across California (United States). 5) The Weather[5] dataset contains 21 meteorological indicators for a range of one year in Germany. 6) the Illness[6] dataset contains the influenza-like illness patients in the United States, the detail of data sets show in Table 6.

Table 6: Summarized feature details of eight datasets.

DATASET	LEN	DIM	FREQ
ETTM2	69680	7	15 MIN
ELECTRICITY	26304	321	1H
EXCHANGE	7588	8	1 DAY
TRAFFIC	17544	862	1H
WEATHER	52696	21	10 MIN
ILI	966	8	7 DAYS

A.2 EXPERIMENT SETTINGS

We used mean squared error (MSE) and mean absolute error (MAE) to evaluate the performance of our model, the most widely used in previous works, for the predict length H,

$$MSE = \frac{1}{H} \sum_{\tau=t}^{t+H} (y_\tau - \hat{y}_\tau)^2 \qquad MAE = \frac{1}{H} \sum_{\tau=t}^{t+H} |y_\tau - \hat{y}_\tau| \qquad (12)$$

The data sets were split into train, validation, and test sets chronologically, following a 70/10/20 split for all the data sets except for ETTm2, which followed a 60/20/20 split (as per convention). Data pre-processing was performed by standardization based on train set statistics.

In the experimental setting, the output lengths were set to 24, 36, 48, and 60 for the ILI data set and 96, 192, 336, and 720 for the other data sets. The input length in all the Transformer models was fixed to 36 for the ILI data set and 96 for the other data sets. N-HiTS and FiLM determine the input length by tuning the parameters and investigating their performance in different data sets. To fairly compare the performance of these models, we used two ways to select the input length: 1) the input length was fixed at 96, denoted as FV-MgNet (96), which has the same settings as the Transformer; and 2) the input length was fixed at one of 96, 192, 336, or 720 for each data set, denoted as FV-MgNet. This latter method can better demonstrate the advantages of FV-MgNet.

A.3 UNIVARIATE FORECASTING RESULTS

We tested the two data sets in a univariate setting and the results are shown in Table 7. Compared with the extensive baselines, FV-MgNet achieves competitive results, especially in long-term forecasting on ETTm2. FV-MgNet can reduce MSE error by 12%.

---

[1] https://github.com/zhouhaoyi/ETDataset

[2] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams 20112014

[3] https://github.com/laiguokun/multivariate-time-series-data

[4] http://pems.dot.ca.gov

[5] https://www.bgc-jena.mpg.de/wetter/

[6] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

Table 7: Univariate forecasting results with different prediction lengths $O \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. The best results are highlighted in bold.

Methods		FV-MgNet		FiLM		N-HiTS		ETSformer		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	0.077	0.213	0.065	0.189	0.066	**0.185**	0.080	0.212	**0.063**	0.189	0.065	0.189
	192	0.101	0.247	0.094	0.233	**0.087**	**0.223**	0.150	0.302	0.102	0.245	0.118	0.256
	336	0.121	0.272	0.121	0.274	**0.106**	**0.251**	0.175	0.334	0.130	0.279	0.154	0.305
	720	**0.153**	**0.310**	0.173	0.323	0.157	0.312	0.224	0.379	0.178	0.325	0.182	0.335
Exchange	96	0.123	0.245	0.110	0.259	**0.093**	**0.223**	0.099	0.230	0.131	0.284	0.241	0.387
	192	0.295	0.402	**0.207**	0.352	0.230	**0.313**	0.223	0.353	0.277	0.420	0.300	0.369
	336	0.337	0.486	**0.327**	**0.461**	0.370	0.486	0.421	0.497	0.426	0.511	0.509	0.524
	720	**0.641**	0.663	0.811	0.708	0.728	**0.569**	1.114	0.807	1.162	0.832	1.260	0.867

A.4 COMPARISON WITH TRANSFORMER MODELS UNDER THE SAME SETTINGS

Transformer-type models set the input length I as 36 for ILI and 96 for the others. We also experimented with FV-MgNet under the same settings, and we called this model FV-MgNet(96); the baseline of Transformer models contain FEDformer (Zhou et al. (2022b)), ETSformer (Woo et al. (2022)), Autofomer (Wu et al. (2021)), Informer (Zhou et al. (2021)), LogTransfomer (Li et al. (2019)), Reformer (Kitaev et al. (2020)), the results are shown in Table 8. Compared with ETSformer, FV-MgNet decreases MSE error by more than 8% and MAE error by 11%. On the Exchange, Traffic and ILI data sets, the improvement was particularly obvious, with decreases in MSE error of 15%, 10%, and 25% respectively. The experimental results show that FV-MgNet can also well capture the information in time series under the short input length.

Table 8: Multivariate results with different prediction lengths $O \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates a better prediction. We set the input length I as 36 for ILI and 96 for the others. The best results are highlighted in bold.

Methods		FV-MgNet(96)		ETSformer		FEDformer		Autoformer		Informer		LogTrans		Reformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	**0.176**	**0.252**	0.189	0.280	0.203	0.287	0.255	0.339	0.365	0.453	0.768	0.642	0.658	0.619
	192	**0.241**	**0.296**	0.253	0.319	0.269	0.328	0.281	0.340	0.533	0.563	0.989	0.757	1.078	0.827
	336	**0.303**	**0.335**	0.314	0.357	0.325	0.366	0.339	0.372	1.363	0.887	1.334	0.872	1.549	0.972
	720	**0.401**	**0.399**	0.414	0.413	0.421	0.415	0.422	0.419	3.379	1.388	3.048	1.328	2.631	1.242
Electricity	96	0.204	**0.287**	0.187	0.304	**0.183**	0.297	0.201	0.317	0.274	0.368	0.258	0.357	0.312	0.402
	192	0.208	**0.291**	0.199	0.315	**0.195**	0.308	0.222	0.334	0.296	0.386	0.266	0.368	0.348	0.433
	336	0.222	**0.306**	0.212	0.329	**0.212**	0.313	0.231	0.338	0.300	0.394	0.280	0.380	0.350	0.433
	720	0.258	**0.332**	0.233	0.345	**0.231**	0.343	0.254	0.361	0.373	0.439	0.283	0.376	0.340	0.420
Exchange	96	**0.082**	0.206	0.085	**0.204**	0.139	0.276	0.197	0.323	0.847	0.752	0.968	0.812	1.065	0.829
	192	0.184	0.314	**0.182**	**0.303**	0.256	0.369	0.300	0.369	1.204	0.895	1.040	0.851	1.188	0.906
	336	**0.307**	**0.416**	0.348	0.428	0.426	0.464	0.509	0.524	1.672	1.036	1.659	1.081	1.357	0.976
	720	**0.554**	**0.582**	1.025	0.774	1.090	0.800	1.447	0.941	2.478	1.310	1.941	1.127	1.510	1.016
Traffic	96	0.564	0.350	0.607	0.392	**0.562**	**0.349**	0.613	0.388	0.719	0.391	0.684	0.384	0.732	0.423
	192	**0.546**	0.345	0.621	0.399	0.562	0.346	0.616	0.382	0.696	0.379	0.685	0.390	0.733	0.420
	336	**0.546**	0.340	0.622	0.396	0.570	**0.323**	0.622	0.337	0.777	0.420	0.733	0.408	0.742	0.420
	720	**0.565**	0.347	0.632	0.396	0.596	0.368	0.660	0.408	0.864	0.472	0.717	0.396	0.755	0.423
Weather	96	**0.185**	**0.219**	0.197	0.281	0.217	0.296	0.266	0.336	0.300	0.384	0.458	0.490	0.689	0.596
	192	**0.231**	**0.256**	0.237	0.312	0.276	0.336	0.307	0.367	0.598	0.544	0.658	0.589	0.752	0.638
	336	**0.283**	**0.256**	0.298	0.353	0.339	0.380	0.359	0.395	0.578	0.523	0.797	0.652	0.639	0.596
	720	0.358	**0.343**	**0.352**	0.388	0.403	0.428	0.419	0.428	1.059	0.741	0.869	0.675	1.130	0.792
ILI	24	**1.648**	**0.805**	2.527	1.020	2.203	0.963	3.483	1.287	5.764	1.677	4.480	1.444	4.400	1.382
	36	**1.894**	**0.837**	2.615	1.007	2.272	0.976	3.103	1.148	4.755	1.467	4.799	1.467	4.783	1.448
	48	**1.962**	**0.856**	2.359.	0.972	2.209	0.981	2.669	1.085	4.763	1.469	4.800	1.468	4.832	1.465
	60	**1.970**	**0.876**	2.487	1.016	2.545	1.061	2.770	1.125	5.264	1.564	5.278	1.560	4.882	1.483

A.5 PARAMETER SENSITIVITY

Here we test the impact of the grid number J and the number of smoothing iterations ν_ℓ on forecasting accuracy. Table 9 shows the impact of grid number when the number of iterations is equal to 2. We set the number of grids as 2, 3, 4, and 5, and observed that in the trade-off between prediction

accuracy and computational efficiency, a grid number of 3 is the best choice. We then explored the number of iterations within the grid on this condition, setting each grid iteration number as 1, 2, 3, 4, and 5. The results in Table 10 show that the optimal number of iterations varies for different datasets, the number of iterations equal to 3 is most suitable for the ETTm2, Exchange, and Weather data sets, and the number of iterations equal to 4 is most suitable for the Electricity and Traffic data sets.

Table 9: The number of grids under each grid iteration is equal to 2; a lower MSE or MAE indicates a better prediction. The best results are highlighted in bold.

Number of grid		2		3		4		5	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	0.244	0.317	0.183	0.268	0.176	0.266	**0.175**	**0.262**
	192	0.263	0.335	0.238	0.313	0.245	0.315	**0.231**	**0.298**
	336	0.336	0.376	**0.283**	**0.334**	0.298	0.350	0.306	0.360
	720	0.465	0.452	**0.378**	**0.397**	0.390	0.400	0.400	0.406
Electricity	96	0.243	0.376	0.154	0.262	**0.149**	**0.253**	0.152	0.256
	192	0.253	0.367	**0.170**	0.274	0.172	0.275	0.166	**0.267**
	336	0.263	0.366	0.182	0.287	**0.178**	**0.280**	0.201	0.301
	720	0.364	0.446	0.224	0.321	**0.223**	**0.318**	0.226	0.322
Exchange	96	0.099	0.230	0.105	0.234	**0.086**	**0.216**	0.105	0.236
	192	0.312	0.423	0.273	0.384	**0.228**	**0.353**	0.196	0.326
	336	0.336	0.446	0.337	0.434	**0.319**	**0.428**	0.332	0.435
	720	1.405	0.924	**0.530**	**0.562**	1.026	0.764	0.980	0.751
Traffic	96	0.524	0.399	**0.419**	**0.293**	0.421	0.297	0.419	0.296
	192	0.545	0.309	**0.431**	**0.303**	0.441	0.306	0.432	0.299
	336	0.550	0.414	**0.446**	**0.305**	0.448	0.307	0.447	0.306
	720	0.578	0.424	**0.470**	**0.311**	0.477	0.318	0.471	0.322
Weather	96	0.180	0.265	**0.158**	0.221	0.159	0.209	0.160	0.209
	192	0.259	0.345	0.205	0.251	**0.200**	**0.246**	0.203	0.251
	336	0.329	0.326	**0.242**	**0.280**	0.246	0.281	0.247	0.292
	720	0.374	0.401	**0.310**	**0.333**	0.315	0.335	0.317	0.342

A.6 INPUT LENGTH

We considered the influence of different input lengths on the forecasting error by setting the input length $I \in \{24, 36, 48, 60\}$ for the ILI data set and $I \in \{96, 192, 336, 720\}$ for the other datasets. The results are shown in Table 11. With the exception of the Exchange dataset, the forecasting error can significantly decrease as the input length increases. This shows that with the increase in length, FV-MgNET can better extract relevant features in the time series and make more accurate forecasting.

A.7 ETT FULL BENCHMARK

We tested our model in the four ETT multivariate data sets and the results are shown in Table 12. ETTh and ETTm have different recording frequencies; the ETTh1 and ETTh2 data sets are recorded hourly, whereas ETTm1 and ETTm2 are recorded every 15 minutes. We obtained comparable results with extensive baseline models and, especially considering the MAE error in the ETTh1 and ETTh2 data sets, FV-MgNet achieves state-of-the-art results.

A.8 RESIDUAL AND \-MGNET ALGORITHM

Table 10: The number of iterations under the grid number is 3; a lower MSE or MAE indicates a better prediction. The best results are highlighted in bold.

Number of iteration		1		2		3		4		5	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	0.176	0.268	0.183	0.268	**0.173**	**0.253**	0.174	0.261	0.175	0.265
	192	0.239	0.313	0.238	0.313	**0.230**	**0.296**	0.230	0.300	0.233	0.301
	336	0.281	0.334	0.283	0.334	**0.279**	**0.329**	0.280	0.329	0.282	0.330
	720	0.386	0.415	0.378	0.415	**0.367**	**0.385**	0.377	0.393	0.379	0.395
Electricity	96	0.145	**0.248**	0.154	0.262	0.161	0.266	**0.144**	0.250	0.202	0.301
	192	0.163	0.265	0.170	0.274	0.166	0.267	**0.163**	**0.262**	0.178	0.279
	336	0.180	0.281	0.182	0.287	0.182	0.284	**0.176**	**0.276**	0.186	0.287
	720	0.221	0.318	0.224	0.321	0.217	0.314	**0.212**	**0.308**	0.216	0.314
Exchange	96	0.084	0.211	0.105	0.234	**0.082**	**0.206**	0.087	0.216	0.083	0.208
	192	0.161	0.295	0.273	0.384	**0.184**	**0.314**	0.208	0.338	0.186	0.315
	336	0.336	0.433	0.337	0.434	**0.307**	**0.416**	0.314	0.427	0.327	0.431
	720	0.996	0.757	**0.530**	**0.562**	0.554	0.582	0.632	0.613	0.916	0.719
Traffic	96	0.413	0.289	0.419	0.293	0.423	0.296	**0.396**	**0.285**	0.429	0.299
	192	0.434	0.301	0.436	0.303	0.435	0.301	**0.417**	**0.295**	0.440	0.304
	336	0.448	0.303	0.446	0.305	0.445	0.304	**0.436**	**0.302**	0.448	0.305
	720	0.472	0.314	0.472	0.318	0.470	**0.313**	**0.468**	0.315	0.472	0.315
Weather	96	0.156	0.206	0.165	0.221	**0.155**	**0.196**	0.160	0.217	0.174	0.265
	192	**0.199**	0.246	0.205	0.251	0.201	**0.239**	0.200	0.251	0.232	0.300
	336	0.247	0.283	0.248	0.284	**0.244**	**0.279**	0.244	0.284	0.280	0.329
	720	0.321	0.351	0.310	0.333	0.313	**0.329**	**0.309**	0.332	0.367	0.385

Table 11: Multivariate long sequence time-series forecasting results on different input lengths. ILI input length $I \in 24, 36, 48, 60$ and others $I \in 96, 192, 336, 720$

Input length		96		192		336		720	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm2	96	0.176	0.252	0.173	0.251	**0.173**	**0.253**	0.180	0.260
	192	0.241	0.296	0.234	**0.291**	**0.230**	0.296	0.249	0.303
	336	0.303	0.335	0.295	0.330	**0.279**	**0.329**	0.304	0.341
	720	0.401	0.393	0.386	0.387	**0.367**	**0.385**	0.398	0.397
Electricity	96	0.204	0.287	0.162	0.259	**0.144**	**0.250**	0.151	0.252
	192	0.208	0.291	0.179	0.274	**0.163**	**0.262**	0.167	0.267
	336	0.222	0.306	0.194	0.289	**0.176**	**0.276**	0.186	0.286
	720	0.258	0.332	0.231	0.316	**0.212**	**0.308**	0.229	0.319
Exchange	96	**0.082**	**0.206**	0.101	0.225	0.119	0.242	0.110	0.240
	192	**0.184**	**0.314**	0.253	0.353	0.251	0.352	0.245	0.366
	336	**0.307**	**0.416**	0.482	0.495	0.435	0.473	0.482	0.503
	720	**0.554**	**0.582**	0.948	0.726	1.219	0.778	1.370	0.830
Traffic	96	0.564	0.350	0.446	0.302	0.414	0.290	**0.396**	**0.285**
	192	0.546	0.345	0.463	0.312	0.435	0.300	**0.417**	**0.295**
	336	0.546	0.340	0.476	0.314	0.449	0.304	**0.436**	**0.302**
	720	0.565	0.347	0.494	0.323	0.473	0.315	**0.468**	**0.315**
Weather	96	0.185	0.219	0.167	0.203	**0.155**	**0.196**	0.153	0.198
	192	0.231	0.256	0.213	0.244	**0.201**	**0.239**	0.203	0.245
	336	0.283	0.296	0.263	0.284	**0.244**	**0.279**	0.256	0.285
	720	0.358	0.343	0.340	0.337	**0.313**	**0.329**	0.330	0.335
Input length		24		36		48		60	
ILI	24	3.487	1.067	1.648	0.805	1.787	0.792	**1.647**	**0.764**
	36	3.785	1.135	1.894	0.837	1.990	0.873	**1.841**	**0.839**
	48	2.487	0.987	1.962	0.856	1.845	0.835	**1.831**	**0.853**
	60	2.387	0.971	1.970	0.876	1.812	0.824	**1.765**	**0.814**

Table 12: Multivariate long sequence time-series forecasting results on ETT full benchmark. The best results are highlighted in bold.

Methods		FV-MgNet		FiLM		FEDformer		Autoformer		Informer		LogTrans		Reformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.364**	**0.382**	0.371	0.394	0.376	0.419	0.449	0.459	0.865	0.713	0.878	0.740	0.837	0.728
	192	**0.409**	**0.415**	0.414	0.423	0.420	0.448	0.500	0.482	1.008	0.792	1.037	0.824	0.923	0.766
	336	0.452	**0.443**	**0.442**	0.445	0.459	0.465	0.521	0.496	1.107	0.809	1.238	0.932	1.097	0.835
	720	0.473	0.482	**0.465**	**0.472**	0.506	0.507	0.514	0.512	1.181	0.865	1.135	0.852	1.257	0.889
ETTh2	96	**0.278**	**0.382**	0.284	0.397	0.346	0.388	0.358	0.397	3.755	1.525	2.116	1.197	2.626	1.317
	192	0.360	**0.436**	0.357	0.452	0.429	0.439	0.456	0.452	5.602	1.931	4.315	1.635	11.12	2.979
	336	**0.365**	**0.457**	0.377	0.486	0.482	0.480	0.482	0.486	4.721	1.835	1.124	1.604	9.323	2.769
	720	**0.418**	**0.432**	0.439	0.456	0.463	0.474	0.515	0.511	3.647	1.625	3.188	1.540	3.874	1.697
ETTm1	96	0.320	0.352	**0.302**	**0.345**	0.378	0.418	0.505	0.475	0.672	0.571	0.600	0.546	0.538	0.528
	192	0.363	0.380	**0.338**	**0.368**	0.426	0.441	0.553	0.496	0.795	0.669	0.837	0.700	0.658	0.592
	336	0.400	0.407	**0.373**	**0.388**	0.445	0.459	0.621	0.537	1.212	0.871	1.124	0.832	0.898	0.721
	720	0.456	0.441	**0.420**	**0.420**	0.543	0.490	0.671	0.561	1.166	0.823	1.153	0.820	1.102	0.841
ETTm2	96	0.173	0.253	**0.165**	**0.256**	0.203	0.287	0.255	0.339	0.365	0.453	0.768	0.642	0.658	0.619
	192	0.240	**0.296**	**0.222**	**0.296**	0.269	0.328	0.281	0.340	0.533	0.563	0.989	0.757	1.078	0.827
	336	0.296	**0.333**	**0.277**	**0.333**	0.325	0.366	0.339	0.372	1.363	0.887	1.334	0.872	1.549	0.972
	720	0.378	**0.385**	**0.371**	0.389	0.421	0.415	0.422	0.419	1.338	3.379	3.048	1.328	2.631	1.242

Algorithm 2 $y = \text{Residual}(f, \nu, I, O)$

1: **Input**: series $f \in \mathbb{R}^I$, number of smoothing iterations ν, input length I, output length O.
2: Initialization: $f^1 = Wf, W \in \mathbb{R}^{I \times I}, u^0 = 0$
3: **for** $i = 1 : \nu$ **do**
4:    Feature extraction (smoothing):

$$u^i = u^{i-1} + \sigma \circ B^i \left(f - Au^{i-1} \right), \quad B^i \in \mathbb{R}^{I \times I}, A \in \mathbb{R}^{I \times I}$$

5: **end for**
   Feature Interpolation:

$$y = W^2 \sigma \left(W^1(u^\nu) + b^1 \right), \quad u^\nu \in \mathbb{R}^I, W_1 \in \mathbb{R}^{I \times O}, W_2 \in \mathbb{R}^{O \times O}$$

Algorithm 3 $y = \backslash -\text{MgNet}(f, J, \nu_\ell, I, O)$

1: **Input**: series $f \in \mathbb{R}^I$, number of grids J, number of smoothing iterations ν_ℓ for $\ell = 1 : J$, input length I, output length O.
2: Initialization: $f^1 = Wf, W \in \mathbb{R}^{I \times I}, u^{1,0} = 0, I_\ell = \frac{I}{2^{\ell-1}}$
3: **for** $\ell = 1 : J$ **do**
4:    **for** $i = 1 : \nu_\ell$ **do**
5:       Feature extraction (smoothing):

$$u^{\ell,i} = u^{\ell,i-1} + \sigma \circ B^{\ell,i} \left(f^\ell - A^\ell u^{\ell,i-1} \right), \quad B^{\ell,i} \in \mathbb{R}^{I_\ell \times I_\ell}, A^\ell \in \mathbb{R}^{I_\ell \times I_\ell}$$

6:    **end for**
7:    Note: $u^\ell = u^{\ell,\nu_\ell}$
8:    **if** $\ell < J$ **then**
9:       Restriction:

$$u^{\ell+1,0} = \Pi_\ell^{\ell+1} u^\ell, \quad \Pi_\ell^{\ell+1} \in \mathbb{R}^{I_{\ell+1} \times I_\ell}$$

$$f^{\ell+1} = R_\ell^{\ell+1}(f^\ell - A^\ell u^\ell) + A^{\ell+1} u^{\ell+1,0}, \quad R_\ell^{\ell+1} \in \mathbb{R}^{I_{\ell+1} \times I_\ell}$$

10:    **end if**
11: **end for**
    Feature Interpolation:

$$y = W^2 \sigma \left(W^1 u^{J,\nu_J} \right), \quad u^{1,\nu_1} \in \mathbb{R}^{I_J}, W_1 \in \mathbb{R}^{I_J \times O}, W_2 \in \mathbb{R}^{0 \times O}$$