A Ablation Experiments

A.1 Hyperparameter Sensitivity

Our model incorporates a hyperparameter K and is selected using the frequencies greater than 10% of the average maximum amplitude across the datasets. In this section, we provide a sensitivity analysis for this parameter and discuss our selection rule of K. As shown in Table 1, the number of dominant frequencies at the 10% threshold contain certain physical significance. Furthermore, the results in Fig. 1 indicate that the 10% threshold typically yields at least sub-optimal results across the datasets. However, accurately selecting the number of dominant frequencies still has room for improvement, which we leave for future work.

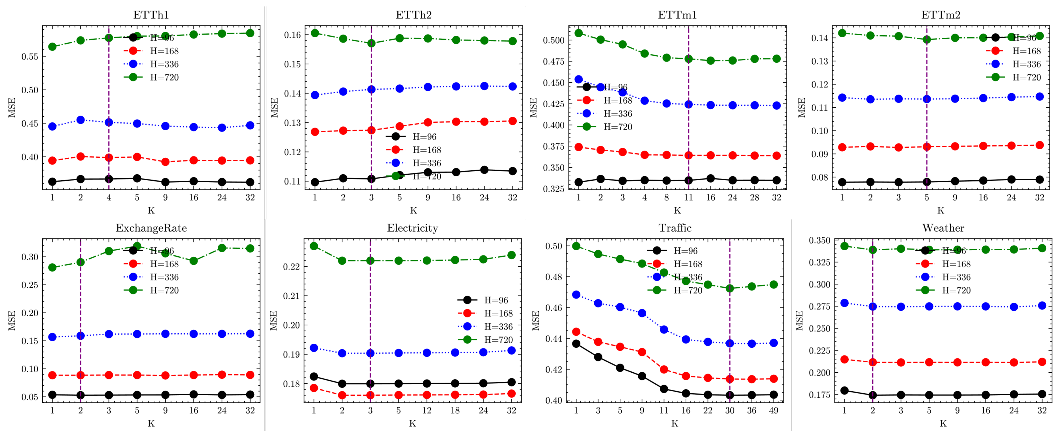

Figure 1: Sensitivity analysis of hyperparameter K, and we select dataset-specific K . We use DLinear as backbone with and use MSE as the evaluation metric, other settings are identical with the main results settings. The purple line denote the selected K in Table 1.

Table 1: selected K by changing the 10% ratio. The green background corresponds to the selected K across our experiments.

ratio	ExchangeRate	Weather	Electricity	Traffic	ETTh1	ETTh2	ETTm1	ETTm2
0	49	49	49	49	49	49	49	49
0.05	3	3	18	49	23	11	11	7
0.1	2	2	3	30	12	5	5	3
0.15	1	1	1	22	8	3	5	2
0.2	1	1	1	16	7	3	3	2
0.25	1	1	1	11	4	1	3	2
0.3	1	1	1	9	3	1	3	1
0.35	1	1	1	5	3	1	3	1
0.4	1	1	1	3	3	1	3	1
0.45	1	1	1	3	2	1	2	1
0.5	1	1	1	2	2	1	2	1
0.55	1	1	1	2	2	1	2	1
0.6	1	1	1	2	2	1	2	1
0.65	1	1	1	2	2	1	2	1
0.7	1	1	1	1	2	1	2	1
0.75	1	1	1	1	2	1	2	1
0.8	1	1	1	1	2	1	2	1
0.85	1	1	1	1	2	1	2	1
0.9	1	1	1	1	2	1	2	1
0.95	1	1	1	1	1	1	2	1
1	0	0	0	0	0	0	0	0

A.2 Addtional Backbones

This section provide addtional backbone results of PatchTST [1], FITS [2], and FreTS [3].

Table 2: Addtional results of PatchTST, FreTS and FITS.

Methods		PatchTST		+FAN		FreTS		+FAN		FITS		+FAN	
Metric		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
ETTm2	96	0.202	0.079	**0.199**	**0.078**	**0.198**	**0.078**	**0.198**	**0.078**	0.204	0.080	**0.198**	**0.078**
	168	0.225	0.097	**0.219**	**0.093**	0.221	0.094	**0.219**	**0.093**	0.221	0.094	**0.219**	**0.093**
	336	**0.239**	**0.112**	0.242	0.114	0.246	0.114	**0.242**	**0.114**	0.244	0.114	**0.241**	**0.113**
	720	0.272	0.142	**0.268**	**0.141**	**0.264**	**0.139**	**0.264**	**0.139**	0.270	0.142	**0.264**	**0.139**
Electricity	96	0.263	0.180	**0.254**	**0.153**	0.274	0.188	**0.272**	**0.185**	0.286	0.202	**0.274**	**0.187**
	168	0.263	0.176	**0.255**	**0.158**	0.273	0.184	**0.273**	**0.182**	0.278	0.188	**0.275**	**0.183**
	336	0.282	0.189	**0.275**	**0.169**	0.292	0.197	**0.292**	**0.195**	0.296	0.200	**0.295**	**0.197**
	720	0.319	0.220	**0.300**	**0.189**	**0.323**	**0.224**	0.325	0.228	0.329	0.233	**0.328**	**0.230**
ExchangeRate	96	0.189	0.063	**0.172**	**0.056**	0.173	0.057	**0.166**	**0.054**	0.168	0.060	**0.167**	**0.054**
	168	0.237	0.102	**0.225**	**0.097**	0.221	0.091	**0.216**	**0.090**	0.219	0.090	**0.217**	**0.088**
	336	0.333	0.198	**0.293**	**0.160**	0.302	0.166	**0.293**	**0.158**	0.308	0.169	**0.293**	**0.159**
	720	0.470	0.355	**0.428**	**0.324**	0.422	0.309	**0.416**	**0.308**	**0.413**	**0.307**	0.417	0.310
Traffic	96	0.323	0.384	**0.314**	**0.374**	0.319	0.387	**0.315**	**0.374**	0.394	0.511	**0.334**	**0.404**
	168	**0.330**	**0.406**	0.334	0.414	0.328	0.405	**0.324**	**0.403**	0.371	0.486	**0.334**	**0.414**
	336	**0.338**	**0.427**	0.340	0.430	0.349	0.429	**0.340**	**0.426**	0.385	0.510	**0.345**	**0.436**
	720	0.378	0.460	**0.373**	**0.454**	0.389	0.465	**0.370**	**0.471**	0.407	0.536	**0.371**	**0.471**
Weather	96	0.222	0.173	**0.220**	**0.170**	0.217	0.175	**0.214**	**0.173**	0.248	0.203	**0.225**	**0.182**
	168	0.257	0.210	**0.251**	**0.209**	**0.250**	**0.211**	0.254	0.210	0.272	0.239	**0.260**	**0.218**
	336	0.305	0.283	**0.301**	**0.278**	**0.292**	**0.271**	0.298	0.275	0.330	0.297	**0.297**	**0.281**
	720	0.363	0.351	**0.350**	**0.344**	**0.337**	**0.333**	0.345	0.340	0.376	0.354	**0.352**	**0.346**

B Dataset Metrics

In this section, we provide a detailed discussion of the calculation methods for certain metrics, aiming to enhance the reproducibility of our work. We provide a Jupyter notebook to reproduce the result, at our public Github repository[1].

B.1 Trend Variation

To capture global trend shifts, we calculate the mean values over different regions of the dataset. Specifically, given a timeseries dataset $\mathcal{X} \in \mathbb{R}^{N \times D}$, we first chronologically split it into $\mathcal{X}^{\text{train}}$, $\mathcal{X}^{\text{val}}$, and $\mathcal{X}^{\text{test}}$, representing the training, validation, and testing datasets, respectively. The trend variations are then computed as follows:

$$\text{Trend Variation} = \left| \frac{\text{Mean}_N(\mathcal{X}^{\text{train}}) - \text{Mean}_N(\mathcal{X}^{\text{val,test}})}{\text{Mean}_N(\mathcal{X}^{\text{train}})} \right| \tag{1}$$

where the subscripts indicate the dimension of mean, $|\cdot|$ denotes the absolute value operation, and $\mathcal{X}^{\text{val,test}}$ represents the concatenation of the validation and test sets. Note that, to obtain relative results across different datasets, the trend variation is normalized by dividing by the mean of the training dataset. We fetch the first dimension to be the value in main content Table 1.

B.2 Seasonal variations.

We evaluate seasonal changes by analyzing the variations in Fourier frequencies across all input instances. Given the inputs, $X \in \mathbb{R}^{N_i \times L \times D}$ where N_i is the number of inputs. We first obtain the FFT results of all inputs, denoted as $Z \in \mathbb{C}^{N_i \times L \times D}$. Then, we calculate the variance across different inputs and normalize this variance by dividing by the mean of each input, computed as:

$$\text{Seasonal Variation} = \frac{\text{Var}_{N_i}[\text{Amp}(Z)]}{\text{Mean}_L(X)} \tag{2}$$

where the subscripts indicate the dimension of the operation. We sum the results across all channels for the value in main text Table 1.

[1] `https://github.com/icannotnamemyself/FAN/blob/main/notebooks/metrics.ipynb`

References

[1] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[2] Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

[3] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.