FragFM: Fragmented Fourier Matrix for multi-seasonality extraction in Univariate Long-Term Time Series Forecasting

Aditya Dey[1 [0000−0002−8086−1243]], Jonas Kusch[1 [0000−0002−2061−2114]], and Fadi Al Machot[1 [0000−0002−1239−9261]]

Department of Data Science, Faculty of Science and Technology, Norwegian University of Life Sciences, P.O. Box 5003, 1432 Ås, Norway
{aditya.dey, jonas.kusch, fadi.al.machot,}@nmbu.no

Abstract. Long-term time series forecasting often relies on frequency- or time-domain decomposition to capture trends and seasonalities. Frequency based methods, such as the Discrete Fourier Transform, isolate dominant periodic components and suppress noise, but they operate in the complex domain. Time-domain models apply trend–seasonality decomposition but often capture only a single seasonality entangled with residual noise. To address these limitations, we propose the Fragmented Fourier Matrix (FragFM)-Fragmented Fourier Matrix–a time domain framework that incorporates spectral insights through fixed Fourier fragments. FragFM removes trends via moving averages, extracts multiple seasonal components with fragmented Fourier bases, and corrects residual noise. Operating fully in the real domain, FragFM achieves forecasting accuracy comparable to leading frequency-domain models while consistently outperforming traditional time-domain approaches. Its lightweight, interpretable design offers an efficient solution for long-term forecasting, bridging time- and frequency-domain methods. The code repository is: https://github.com/KODX-DV1NUX/FragFM (cf. the abstract).

Keywords: Long Term Time Series Forecasting · Univariate Forecasting · Low Rank · Fourier Transform · Seasonal Decomposition.

1 Introduction

Time series forecasting has evolved from short-term, point-based predictions, where models estimate only the next time step based on a given input window using statistical or machine learning approaches, to long-horizon forecasting, often referred to as Long-term Time Series Forecasting (LTSF), covering future ranges of 24 to 720 steps or more. While point forecasting provides insight into immediate future values, LTSF offers a broader and more strategic perspective on system behaviour and evolution. Such extended prediction horizons enable more informed decision-making across diverse domains: in economics for trend

analysis [9], in finance for risk management and portfolio planning [5], and in industrial machinery for enabling predictive maintenance [1].

The field of LTSF has seen rapid progress, with Transformer models becoming popular for their ability to process raw input sequences using self-attention [8], capturing both short- and long-term dependencies and often reflecting underlying seasonal and trend structures in time series. Also, Multi-layer Perceptron (MLP)-based models have shown equivalent performance, with or without requiring explicit decomposition into trend and seasonal components [14].

However, these time-domain models are often outperformed by frequency-domain approaches [17,12]. Spectral representation using Discrete Fourier Transform (DFT) [11] decomposes the signal into distinct frequency bands. High-frequency components, typically associated with noise, can be suppressed, while dominant low-frequency patterns are retained and effectively learned [12]. This advantage has contributed to the strong empirical performance of frequency-based models [15,10].

A key limitation of traditional time-domain decomposition methods is their reliance on a single seasonal component [14,4], which is inadequate, as real-world time series often exhibit multiple overlapping seasonalities—such as daily, weekly, and yearly cycles. Additionally, noise usually remains entangled within seasonal components, limiting their effectiveness.

To overcome the limitations of traditional time-domain methods, we propose FragFM, a decomposition model for multi-seasonality extraction and noise reduction. It first isolates the trend using a moving average filter, then extracts seasonal components via symmetric fragmented Fourier matrices—constructed from fixed Fourier bases and trainable real-valued eigenvalues. Spectral symmetry is enforced to ensure real-valued outputs. By partitioning the Fourier basis into localised fragments across frequency bands, FragFM efficiently captures diverse periodic patterns in the time domain. Multiple decomposition blocks progressively remove residual noise, and the final prediction is performed through a low-rank layer, maintaining interpretability with minimal parameter overhead.

Since LTSF models adopt a channel-independent approach to multivariate series [14,8,6], treating each series as a separate univariate forecasting task, univariate comparisons provide a clear measure of a model's capabilities. Under this evaluation, FragFM demonstrates competitive performance among leading frequency-domain models while maintaining a lightweight architecture and consistently outperforming traditional time-domain approaches.

2 Related Works

Temporal-domain models like PatchTST [8] and iTransformer [6] operate without explicit decomposition; instead, they modify the input embedding to capture underlying relationships. Convolutional and Recurrent networks, such as ModernTCN [7] and WiTRAN [2], do not explicitly decompose either, as the design extracts both long- and short-term dependencies, thus implicitly capturing re-

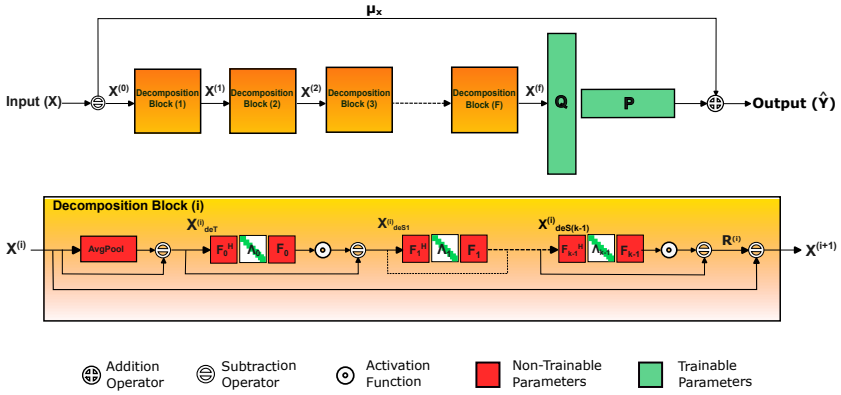


Fig. 1. Architecture of Fragmented Fourier Matrix (FragFM) for extracting multiple seasonalities to isolate residual noise per decomposition block cleaning the signal iteratively, followed by direct prediction strategy using low-rank layer.

lations. In contrast, MLP-based models like DLinear [14] benefit from explicit temporal decomposition into trends and residuals.

Frequency-domain models, such as Fedformer [17], apply self-attention in the spectral domain, leveraging frequency representations of time series. Similarly, FrNet [15] introduces frequency-aware rotation networks that, when combined with self-attention, allow for capturing phase information. FITS [12] an MLP model; apply DFT and frequency cut-off methods to reduce the effects of high-frequency noise and predict using low-frequency components.

These spectral-domain models offer better noise-reduction techniques but face an inherent limitation. A high-frequency resolution provides aggregated contributions from all time points, representing trend and seasonality patterns at the cost of low-temporal resolution, thereby losing information about when specific components occur [11]. Therefore, a trade-off exists between the resolution that identifies contributing time points and the noise components.

Hence, our attempt at using multi-seasonality extraction, which leverages spectral properties within temporal domain frameworks, facilitates better separation of noise components and provides a balanced trade-off that may allow the temporal models to achieve similar accuracies as spectral models.

3 Proposed Method

Problem Definition: Given a regularly sampled univariate time series $X = \{x(0), x(1), \ldots, x(L)\} \in \mathbb{R}^L$ with a lookback window of length L, the goal is to forecast future values $\hat{Y} = \{x(L+1), \ldots, x(L+H)\} \in \mathbb{R}^H$ over a prediction horizon of length H.

Summary: Given a normalised input $X^{(0)}$, we iteratively remove residual noise using decomposition blocks ($i = \{0, \ldots, f-1\}$). Each decomposition block (i)

takes an input $X^{(i)}$, performs de-trending to obtain Trend ($X^{(i)}_{\text{deT}}$), and then performs de-seasonalization ($X^{(i)}_{\text{deS}_j}$) on the Trend by removing k seasonal components. This process leads to a residual ($R^{(i)}$) that is removed from the input to clean the signal. The detailed description is provided in Section 3.1.

$$X^{(0)} = X - \mu_X \qquad \text{(Input Normalization)}$$

For $i = 0$ to $f - 1$:

$$R^{(i)} = X^{(i)} - X^{(i)}_{\text{deT}} - \sum_{j=0}^{k-1} X^{(i)}_{\text{deS}_j} \qquad \text{(Residual Computation)}$$

$$X^{(i+1)} = X^{(i)} - R^{(i)} \qquad \text{(Residual Subtraction)} \tag{1}$$

The last block's output ($X^{(f)}$) is forwarded to a low-rank linear layer, where the weight matrix is factorised as $PQ \in \mathbb{R}^{H \times L}$, with rank $r \ll \min(L, H)$, to generate the forecast $\hat{Y}$ with the input mean μ_X added back to restore the original scale.

$$\hat{Y} = PQX^{(\text{f})} + B + \mu_X \qquad \text{(Forecasting)} \tag{2}$$

3.1 Decomposition Block

In this section, we describe in detail the de-trending and de-seasonalizing processes within a single decomposition block (i) and explain how they facilitate the isolation of noise.

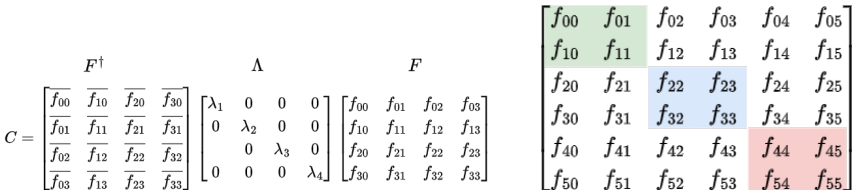
De-trending process: We employ an average pooling operation with a kernel size k and a stride of 1, along with appropriate padding to preserve the input length to extract the trend component, following prior work [14]. Subtracting this estimated trend from the original signal yields the de-trended signal ($X^{(i)}_{\text{deT}}$).

$$X^{(i)}_{\text{deT}} = X^{(i)} - \text{AvgPool}(X^{(i)}) \tag{3}$$

De-seasonalizing process: After de-trending, the next step involves identifying seasonality. Seasonal components in time series typically correspond to repeating cyclic patterns. This cyclic behaviour can be effectively modelled using a circulant matrix, a structured matrix in which each row is a cyclic right-shift of the previous row [3].

Real-valued circulant matrices possess a unique property: their eigendecomposition is expressed as $C = F^{\dagger}\Lambda F$, where F is the unitary Fourier matrix, $F^{\dagger}$ denotes its Hermitian transpose, and $\Lambda = \text{diag}(\lambda_0, \ldots, \lambda_{n-1})$ contains the real-valued eigenvalues (Figure 2).

If one constructs a real-valued circulant matrix from this eigen-decomposed form–keeping the eigenvectors fixed while learning the eigenvalues—there is no inherent guarantee that the resulting matrix will remain real-valued. To ensure real-valued outputs, symmetry must be enforced on the eigenvalue matrix ($\Lambda = \Lambda^T$) , which in turn yields a symmetric circulant matrix ($C = C^T$). This



(a) Circulant Matrix (b) Fragmented Fourier Matrix

Fig. 2. (a) An example of eigendecomposition of 4×4 circulant matrix. (b) An example of Fragmented Fourier Matrix of 6×6, which is divided into 3 fragments.

symmetry induces bidirectional behaviour, allowing each time point to interact with both past and future lags–a property resembling bidirectional Recurrent Neural Network (RNN)s.

However, creating many such matrices with trainable eigenvalues to capture multiple seasonalities introduces redundancy, as they share the same set of Fourier eigenvectors spanning an overlapping spectral region. To mitigate this, we propose Fourier Fragments–fixed, non-overlapping spectral partitions derived from a larger unitary Fourier matrix.

Given a large unitary Fourier matrix $F \in \mathbb{C}^{kL \times kL}$, where k denotes the number of fragments and L the lookback window length, we define blocks $F_p = F[pL : (p+1)L, pL : (p+1)L]$ as shown in Figure 3.1. Here, $p = \{0, \ldots, k-1\}$ is the hyperparameter for the number of seasonalities. The individual entries of each fragment are given by:

$$(F_p)_{r,s} = \frac{1}{\sqrt{kL}} \omega^{(pL+r)(pL+s)}, \quad (\text{where } \omega = e^{-\frac{2\pi i}{kL}})$$

Each fragment acts as a localised spectral filter: the term $\omega^{p^2L^2}$ centres on a frequency band, ω^{rs} captures within-block interactions, and $\omega^{pL(r+s)}$ ensures distinct periodic components across fragments. Thus, each fragment specialises in a distinct frequency band, enabling efficient and interpretable multi-scale seasonal extraction.

Using these fragmented Fourier matrices F_p as fixed eigenvectors and a trainable, symmetric eigenvalue matrix Λ_p, we define a trainable seasonal matrix $A_p = F_p^H \Lambda_p F_p$. Each matrix corresponds to a specific seasonal frequency band and progressively removes its contribution from the signal:

$$X^{(i)}_{deS_{j+1}} = X^{(i)}_{deS_j} - \phi\left(A^{(i)}_j X^{(i)}_{deS_j}\right), \qquad j = 0, \ldots, k-1, \tag{4}$$

where $\phi(\cdot)$ denotes a bounded nonlinearity, such as the hyperbolic tangent and sigmoid. We use spectral regularisation to ensure approximate symmetry, as shown in equation (5).

$$\mathcal{L}_{\text{sym}} = \sum_{i=0}^{f-1} \sum_{p=0}^{k-1} \|A^{(i)}_p - (A^{(i)}_p)^T\|_F^2. \tag{5}$$

The iterative extraction of these seasonal components is able to de-seasonalize the signal, yielding the final residual $R^{(i)}$, as summarised in equation 1. The residual component is subtracted from the original input to obtain a de-noised (cleaned) signal as the output of the decomposition block.

4 Results

We conduct extensive experiments on six real-world time series datasets commonly used for LTSF benchmarking: ETT, Electricity, and Traffic [16]. For the ETT datasets, we adopt a $60 : 20 : 20$ split for training, validation, and testing, and a $70 : 10 : 20$ split for the remaining datasets. The comparison covers a broad spectrum of baselines spanning both temporal-domain methods—DLinear [14], SparseTSF [4], iTransformer [6], and PatchTST [8] and spectral-domain approaches—FITS [12], FreTS [13], and FrNet [15].

All models are evaluated on univariate forecasting tasks with input lengths $L \in 336, 512, 720$ and prediction horizons H ranging from 48 to 720. For FragFM, the configuration for ETT employs two decomposition blocks with $p = 2$ seasonal components, a trend kernel size of $k = 70$, and a low-rank approximation of $r = 25$. For the Traffic and Electricity datasets, a single decomposition block is used with $p = 3$ seasonal components, a trend kernel size of $k = 30$, and a rank of $r = 40$.

Forecasting accuracy is evaluated using the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics. The results reported in Table 1 correspond to the input length that yields the lowest MSE for each prediction horizon. We also report the total number of trainable parameters and FLOPs (floating-point operations) for all models, computed using a batch size of 128, an input length $L = 512$, a prediction horizon $H = 192$, and each model's optimal configuration. The FLOPs are estimated based on the number of operations in the forward propagation.

The results indicate that FragFM achieves state-of-the-art performance across the ETT datasets, with the exception of ETTh2, where iTransformer slightly outperforms it; though this performance can be considered an outlier, given iTransformer's limited generalisation elsewhere. FragFM attains accuracy comparable to or superior to leading baselines such as FITS and DLinear, which represent spectral and time-domain models, respectively. Notably, this accuracy is achieved with only 19.64K trainable parameters, corresponding to approximately 93% reduction in parameter space relative to DLinear (the best time-domain model) and about 29% reduction compared to FITS (the best spectral-domain model). It should be noted that complex-valued parameters consist of real and imaginary components, counted as single entities. Accounting for this, the effective parameter compression is approximately 64% compared to FITS. Although FragFM employs fewer trainable parameters than competing models, its FLOPs are higher due to the increased number of computations per forward pass. This indicates that, while FragFM is more storage-efficient, it requires higher runtime memory and computational throughput.

Table 1. Univariate forecasting results (MSE, MAE; lower is better) on benchmark datasets evaluated with prediction length H between 48–720. The first and second best are represented as **bold** and underline respectively. Total trainable parameters and FLOPs are reported for batch size 128, $L = 512$ and $H = 192$.

		Time-domain										Frequency-domain					
		FragFM		SparseTSF (2024)		iTransformer (2024)		PatchTST (2023)		DLinear (2023)		FrNet (2024)		FreTS (2024)		FITS (2024)	
Dataset	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	48	0.039	0.151	0.049	0.173	0.059	0.192	0.039	0.152	0.039	0.152	**0.038**	**0.150**	0.123	0.274	0.039	0.152
	96	0.054	0.180	0.061	0.194	0.062	0.194	0.054	0.179	0.054	0.177	**0.052**	**0.175**	0.162	0.320	0.054	0.179
	192	0.069	0.205	0.077	0.218	0.069	0.205	0.071	0.206	0.071	0.209	**0.068**	**0.205**	0.186	0.343	0.075	0.215
	336	0.079	0.224	0.088	0.237	**0.074**	**0.217**	0.081	0.229	0.095	0.241	0.079	0.225	0.148	0.306	0.085	0.233
	720	0.078	0.224	0.104	0.253	0.077	0.222	0.089	0.237	0.089	0.237	**0.077**	**0.222**	0.225	0.383	0.103	0.252
ETTh2	48	0.098	0.241	0.096	0.118	0.126	0.277	0.103	0.249	**0.094**	**0.234**	0.095	0.236	0.178	0.327	0.096	0.238
	96	0.133	0.283	0.147	0.302	0.149	0.306	0.134	0.288	0.131	**0.279**	**0.129**	0.280	0.224	0.373	0.132	0.282
	192	0.170	0.326	0.177	0.336	**0.158**	0.324	0.172	0.331	0.174	0.327	0.164	**0.322**	0.261	0.407	0.172	0.329
	336	0.181	0.343	0.187	0.350	**0.162**	**0.329**	0.184	0.347	0.194	0.356	0.170	0.335	0.269	0.421	0.192	0.353
	720	0.216	0.372	0.242	0.398	**0.160**	**0.327**	0.223	0.379	0.252	0.404	0.214	0.372	0.407	0.519	0.240	0.393
ETTm1	48	**0.016**	**0.095**	0.021	0.113	0.020	0.106	0.016	0.095	0.016	0.095	0.016	0.096	0.074	0.206	0.016	0.097
	96	**0.025**	**0.121**	0.029	0.133	0.031	0.132	0.026	0.122	0.026	0.121	0.026	0.122	0.196	0.349	0.026	0.122
	192	**0.038**	**0.148**	0.041	0.156	0.045	0.169	0.040	0.153	0.040	0.150	0.039	0.150	0.148	0.302	0.038	0.149
	336	**0.051**	**0.172**	0.054	0.177	0.054	0.178	0.053	0.175	0.055	0.174	0.051	0.172	0.122	0.276	0.051	0.172
	720	**0.068**	**0.201**	0.071	0.204	0.072	0.206	0.071	0.204	0.071	0.207	0.072	0.205	0.142	0.294	0.070	0.202
ETTm2	48	0.045	0.149	0.064	0.193	0.057	0.172	**0.043**	**0.145**	0.044	0.145	0.046	0.151	0.148	0.303	0.045	0.150
	96	0.063	0.184	0.074	0.208	0.095	0.237	0.064	0.184	**0.063**	**0.182**	0.064	0.188	0.192	0.338	0.063	0.185
	192	**0.090**	**0.225**	0.102	0.247	0.117	0.261	0.095	0.234	0.091	0.226	0.093	0.230	0.167	0.317	0.091	0.226
	336	0.118	0.261	0.127	0.275	0.128	0.279	0.120	0.265	0.118	**0.258**	0.120	0.264	0.161	0.314	**0.117**	0.259
	720	**0.170**	**0.318**	0.176	0.329	0.180	0.333	0.172	0.322	0.174	0.320	0.174	0.328	0.255	0.403	0.170	0.318
Electricity	48	0.166	0.289	0.173	0.293	0.195	0.320	0.169	0.285	**0.158**	**0.278**	0.163	0.281	0.287	0.409	0.159	0.279
	96	0.200	0.313	0.203	0.311	0.219	0.333	0.220	0.319	**0.192**	**0.304**	0.198	0.309	0.340	0.439	0.195	0.306
	192	0.233	0.338	0.234	0.332	0.253	0.356	0.283	0.359	**0.225**	**0.327**	0.245	0.341	0.380	0.455	0.228	0.329
	336	0.269	0.365	0.267	0.359	0.301	0.392	0.317	0.409	**0.257**	0.353	0.282	0.370	0.394	0.467	0.266	0.361
	720	0.316	0.416	0.319	0.413	0.315	0.420	0.364	0.456	**0.290**	**0.393**	0.359	0.448	0.409	0.481	0.310	0.409
Traffic	48	0.126	0.217	0.112	0.180	0.122	0.209	0.113	0.192	**0.107**	**0.179**	0.125	0.213	1.791	0.702	0.108	0.180
	96	0.138	0.233	0.116	0.184	0.149	0.249	0.119	0.198	**0.113**	**0.185**	0.115	0.191	2.438	0.832	0.114	0.188
	192	0.131	0.222	0.117	**0.185**	0.171	0.278	0.122	0.201	**0.116**	0.189	0.116	0.190	0.897	0.547	0.116	0.190
	336	0.129	0.222	0.115	**0.187**	0.144	0.244	0.122	0.205	0.115	0.195	0.136	0.230	1.267	0.700	**0.114**	0.192
	720	0.143	0.236	0.132	**0.207**	0.170	0.278	0.140	0.226	0.133	0.217	0.143	0.239	0.484	0.485	**0.131**	0.212
Parameters		19.64K		0.40K		972.73K		214.27K		295.48K		196.89K		16892.99K		27.88K	
FLOPS		1.21×10^9		0.475×10^3		0.626×10^9		0.207×10^9		0.025×10^9		0.036×10^9		2.162×10^9		0.003×10^9	

On the Electricity and Traffic datasets, FragFM underperforms compared to FITS and DLinear. The strong performance of DLinear stems from its decomposed prediction strategy, where trend and residual components are forecasted separately–a design choice that could potentially be adapted within FragFM to enhance its predictive accuracy.

In summary, FragFM demonstrates robust forecasting capability by integrating fragmented Fourier matrices, trainable eigenvalues, and low-rank prediction layers. Its improvements arise from effectively capturing multiple seasonalities and iteratively removing residual noise, enabling competitive performance across diverse long-term forecasting benchmarks.

5 Ablation Studies

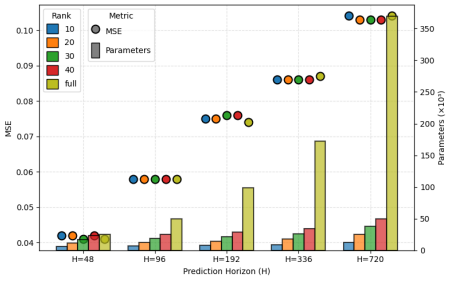


Fig. 3. Comparison of full-rank (W) and low-rank (PQ) on the ETTh1 dataset with input length $L = 512$, showing their MSE performance and total trainable parameters.

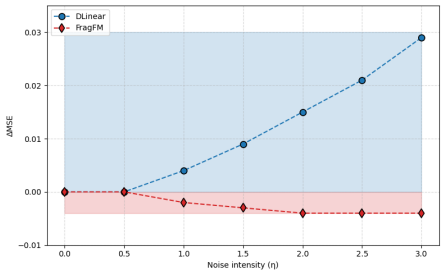


Fig. 4. Noise robustness of FragFM and DLinear for $L = 336$ and $H = 512$, showing ΔMSE across increasing noise intensities. Shaded regions indicate deviation from $\eta = 0.0$.

Parameter compression using low-rank: We compare low-rank (PQ) approximations using ranks $r \in \{10, 20, 30, 40\}$ in equation (2) by replacing it with full-rank (W), i.e., the standard linear layer. The rest of the configuration is a trend kernel size $k = 70$, one decomposition block, and a $p = 1$ seasonal component, tested on the ETTh1 dataset. As shown in Figure 3, the total number of trainable parameters in the full-rank model increases rapidly with prediction length, whereas low-rank models achieve comparable MSE accuracy while exhibiting approximately linear parameter growth. This indicates that much of the parameter capacity in full-rank models is redundant and can be replaced with low-rank approximations, reducing the FLOPs required for the final layer prediction and freeing computational resources for learning within the decomposition blocks.

Noise Robustness: We evaluate noise robustness by adding scaled Gaussian noise to the training data: $X_{\text{train}} := X_{\text{train}} + \eta \times \mathcal{N}(0, 1)$, where $\eta \in \{0.0, 0.5, 1.5, 2.0, 3.0\}$

represents increasing noise intensity from none to extreme. For FragFM, we use 2 decomposition blocks with 2 seasonal components, rank $r = 30$, and kernel size $k = 70$, and we compare against DLinear (the best time-domain model); both models are tested with $L = 336$ and $H = 512$. Figure 4 plots noise intensity versus $\Delta\text{MSE} = \text{MSE}_{\eta=x} - \text{MSE}_{\eta=0.0}$, with shaded regions indicating the maximum deviation from $\eta = 0.0$. The results show that even under extreme noise, FragFM exhibits substantially lower deviation compared to DLinear; the maximum deviation for FragFM is 0.004, while DLinear reaches 0.03. This demonstrates that FragFM's decomposition blocks, combined with spectral regularisation and low-rank layers acting as implicit regularizers, provide strong resistance to noise.

6 Conclusion

FragFM demonstrates that integrating time- and frequency-domain decomposition via fragmented Fourier matrices can achieve state-of-the-art forecasting accuracy. Its underperformance on the Electricity and Traffic datasets, compared to DLinear, suggests that prediction strategies leveraging decomposed trend and residual components may offer further advantages and warrant additional exploration. Moreover, a deeper analysis of the behaviour induced by fragmented Fourier matrices is needed, particularly since operations on Circulant matrices could be efficiently implemented via circular convolution techniques.

In summary, FragFM leverages a symmetric fragmented Fourier matrix with trainable eigenvalues to efficiently capture multiple seasonalities and suppress noise. Across diverse benchmarks, it consistently achieves competitive or superior results, demonstrating the potential of structured Fourier fragments as a lightweight, interpretable, and effective framework for time-domain forecasting. Its strong noise resilience arises from spectral regularisation within the fragmented Fourier matrices, while the low-rank prediction layer provides both parameter compression and additional regularisation, making FragFM robust and well-suited for real-world applications.

Acknowledgments. The co-author F.A.M. acknowledges funding from the EC's Horizon Europe research and innovation programme within the project BatCAT, Battery Cell Assembly Twin, under grant agreement no. 101137725.

Disclosure of Interests. There are no competing interests to be disclosed.

References

1. Chen, C., Fu, H., Zheng, Y., Tao, F., Liu, Y.: The advance of digital twin for predictive maintenance: The role and function of machine learning. Journal of Manufacturing Systems **71**, 581–594 (2023)
2. Jia, Y., Lin, Y., Hao, X., Lin, Y., Guo, S., Wan, H.: Witran: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting. Advances in Neural Information Processing Systems **36** (2024)

3. Karner, H., Schneid, J., Ueberhuber, C.W.: Spectral decomposition of real circulant matrices. Linear algebra and its applications **367**, 301–311 (2003)
4. Lin, S., Lin, W., Wu, W., Chen, H., Yang, J.: Sparsetsf: Modeling long-term time series forecasting with* 1k* parameters. In: International Conference on Machine Learning. pp. 30211–30226. PMLR (2024)
5. Liu, X.Y., Yang, H., Gao, J., Wang, C.D.: Finrl: Deep reinforcement learning framework to automate trading in quantitative finance. In: Proceedings of the second ACM international conference on AI in finance. pp. 1–9 (2021)
6. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=JePfAI8fah
7. Luo, D., Wang, X.: Moderntcn: A modern pure convolution structure for general time series analysis. In: The twelfth international conference on learning representations. pp. 1–43 (2024)
8. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=Jbdc0vTOcol
9. Poledna, S., Miess, M.G., Hommes, C., Rabitsch, K.: Economic forecasting with an agent-based model. European Economic Review **151**, 104306 (2023)
10. Shen, R., Liu, L., Wang, B., Guan, Y., Yang, Y., Jiang, J.: Freqtsf: Time series forecasting via simulating frequency kramer-kronig relations. arXiv e-prints pp. arXiv–2407 (2024)
11. Sundararajan, D.: The discrete Fourier transform: theory, algorithms and applications. World Scientific (2001)
12. Xu, Z., Zeng, A., Xu, Q.: FITS: Modeling time series with $10k$ parameters. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=bWcnvZ3qMb
13. Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., Niu, Z.: Frequency-domain mlps are more effective learners in time series forecasting. Advances in Neural Information Processing Systems **36** (2024)
14. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
15. Zhang, X., Feng, S., Ma, J., Lin, H., Li, X., Ye, Y., Li, F., Ong, Y.S.: Frnet: Frequency-based rotation network for long-term time series forecasting. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3586–3597 (2024)
16. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 11106–11115 (2021)
17. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning. pp. 27268–27286. PMLR (2022)