TIME-SERIES FORECASTING MODELS CARE LESS ABOUT WHAT HUMANS SEE

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting predicts future values from historical observations to support energy scheduling, weather prediction, and traffic management. In recent years, Fourier operations have been widely used in time-series forecasting models. Their reported gains are widely attributed to the explicit representation of periodic structure with a prescribed Fourier basis. Fourier operations commonly introduce multiple filtered views of the same history, but existing studies have not analyzed whether the reported gains come from the Fourier basis or the accompanying view expansion. We examine this attribution with matched Random controls. We replace Fourier filters or bases with norm- and parameter-matched Random counterparts. We conduct 96 matched comparisons across generic backbones, backbone scales, and eight published models, covering input histories with varying degrees of periodic repetition. Fourier shows no aggregate forecasting advantage over Random, with 47 Fourier wins, 47 Random wins, and two ties. Both representations often improve over Raw. Further analysis shows that their shared gain primarily reflects expanding each history into multiple distinct filtered views. Increasing the response count usually improves forecasting, whereas longer filters provide no corresponding benefit. Prescribed frequency structure can provide additional gains when it matches repeated patterns within the input window.

1 INTRODUCTION

Time-series forecasts support decision making in energy, weather, and transportation (Lim & Zohren, 2021; Yi et al., 2023b). Forecasting models estimate future values by learning temporal dependencies from an observed history. Fourier bases express temporal variation as sinusoidal components at different frequencies. In traditional signal analysis, their frequencies, amplitudes, and phases help people interpret the periodic structure of a sequence. This interpretability provides an intuitive motivation for using Fourier representations in time-series forecasting. Fourier operations have become common in forecasting models that use spectral coefficients, dominant periods, and spectral filters (Zhou et al., 2022; Wu et al., 2023; Xu et al., 2024; Yi et al., 2023b;a; Yang et al., 2024; Yi et al., 2024; Huang et al., 2025; Liu, 2025; Zhang et al., 2026; An et al., 2026). Their reported gains are commonly attributed to periodicity, global temporal dependence, spectral energy concentration, or the separation of frequency components.

The interpretability of a Fourier basis does not establish why a forecasting model benefits from using it. A Fourier filter bank transforms one history into several filter responses and presents these filtered views jointly to the forecasting model. The resulting input contains both prescribed frequency structure and an expanded set of views. A comparison with the untransformed history measures their combined effect, without determining which change accounts for the gain. Separating these changes allows us to examine whether the forecasting benefit requires a Fourier basis.

We construct Random representations using fixed random filters that match the number and scale of the Fourier responses without encoding a Fourier basis. Fixed random transformations can provide effective representations, as demonstrated by random features, random projection filter banks, and random convolutional kernels (Rahimi & Recht, 2007; Farahmand et al., 2017; Dempster et al., 2020). Within each matched comparison, both representations use the same forecasting model and training protocol. We conduct 96 matched comparisons across generic backbones, backbone scales, and eight published Fourier-based forecasting models. These comparisons span input histories with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

varying degrees of periodic repetition. Fourier shows no aggregate forecasting advantage over Random, but can provide additional gains when its prescribed frequency structure matches repeated patterns within the input window. Random filters retain comparable forecasting performance without the prescribed frequency relations that make Fourier representations interpretable.

Both representations often achieve lower forecasting error than Raw, the untransformed history. Their comparable performance motivates an analysis of the changes they share. We separate changes in the input metric from the component that produces distinct filter responses. Repeated Raw views provide a comparison that increases the number of views without introducing different filter responses. The analysis identifies multiple distinct filtered views as the primary source of the gain shared by Fourier and Random.

Our contributions are threefold.

1. We introduce a controlled attribution framework to distinguish prescribed frequency structure from view expansion. Matched comparisons show no aggregate forecasting advantage for Fourier over Random.
2. We identify multiple distinct filtered views of the same history as the main source of the gain shared by Fourier and Random. This explains why random filters can retain forecasting benefits without encoding a Fourier basis.
3. We find that more filter responses usually improve forecasting, whereas longer filters provide no corresponding benefit. Prescribed frequency structure can offer additional gains when it matches repeated patterns within the input window. These findings motivate learned or data-adaptive bases that match the temporal structure of the input.

2 Related Work

2.1 Fourier-Based Time-Series Forecasting

Fourier-based forecasters use spectral computation in several ways. Some sample or interpolate compact frequency coefficients for long-range prediction (Zhou et al., 2022; Xu et al., 2024). Others apply complex temporal and cross-variable mappings, learn spectral filters, balance frequency bands, or optimize forecasts in the frequency domain (Yi et al., 2023b;a; 2024; Dayag et al., 2026; Piao et al., 2024; Wang et al., 2025a). Recent models combine spectral transforms with representation learning, frequency decomposition, mixture routing, cycle modeling, or multiresolution temporal patterns (Eldele et al., 2024; Huang et al., 2025; Liu, 2025; Zhang et al., 2026; An et al., 2026; Wang et al., 2025b). These methods motivate Fourier representations through periodicity, global temporal dependence, spectral energy concentration, and decomposition of temporal patterns.

TimesNet and Fourier Basis Mapping make the relation between frequency structure and predictor input especially explicit. TimesNet estimates dominant periods from Fourier amplitudes and reshapes one sequence into several two-dimensional representations associated with those periods (Wu et al., 2023). Fourier Basis Mapping combines Fourier coefficients with their sinusoidal basis functions to construct trajectories that preserve temporal and frequency relations (Yang et al., 2024). In both cases, the prescribed frequency structure is introduced together with a new organization of the predictor input. Comparisons with Raw therefore measure the complete representation rather than the isolated contribution of the Fourier basis. BasisFormer instead learns dataset-level bases, while OLinear constructs a data-adaptive orthogonal transform from temporal correlations (Ni et al., 2023; Yue et al., 2025).

2.2 Transformed and Random Representations

Fixed transformations can change which functions a neural network learns efficiently, even when the transformation itself is not learned from data. Fourier feature mappings reshape the effective spectrum of coordinate networks and enable them to fit higher-frequency functions (Tancik et al., 2020). FNet uses an unparameterized Fourier transform as a token-mixing operator, showing that a fixed transform can support effective sequence modeling (Lee-Thorp et al., 2022). In time-series forecasting, Toner & Darlow (2024) show that several linear forecasting designs are functionally equivalent to unconstrained linear regression over suitably augmented features.

Random representations show that fixed, untrained mappings can construct effective features. Random Fourier features approximate shift-invariant kernels through fixed randomized coordinates (Rahimi & Recht, 2007). Random Projection Filter Bank applies stable random autoregressive filters to time series, and ROCKET constructs time-series features with large collections of random convolutional kernels (Farahmand et al., 2017; Dempster et al., 2020). RPMixer combines random spatial projections with Fourier temporal mixing in separate modules that model spatial and temporal relations (Yeh et al., 2024). Frame projections generate multiple time-series views from orthonormal bases and overcomplete frames (Demirel & Holz, 2025).

3 Controlled Comparison

Identifying the contribution of the Fourier basis requires Fourier and Random inputs to differ only in their filter directions. We define a common filter-bank interface with the same response count and corresponding filter norms. Polar decomposition and the Repeat condition separate changes in the input metric, the number of response branches, and differences among filter responses.

3.1 Matched Fourier and Random Inputs

Let $X \in \mathbb{R}^{L\times C}$ denote a multivariate history with L time points and C variables. The same temporal filter bank is applied independently to every variable. Each filter has length P and produces one filter response of length L. Each filter response forms one filtered view of the history and is processed by the shared predictor as one response branch. The response count K equals the number of filters in the bank. For a single-variable history $x \in \mathbb{R}^L$, the bank defines the linear map

$$x \in \mathbb{R}^L, \qquad Bx \in \mathbb{R}^{L\times K}, \qquad Bx = [b_1 * x, \ldots, b_K * x], \qquad Z_B(x) = [x, Bx].$$

Here $*$ denotes unit-stride circular filtering, and each column of Bx is one filter response. The expanded input $Z_B(x)$ retains the unchanged sequence as the Raw branch and appends the K response branches. We refer to this operation as view expansion throughout the paper. The predictor processes every branch with shared parameters, and a linear readout combines their forecasts. The fixed bank adds no trainable parameters. Increasing K increases the number of predictor evaluations and the width of the readout while leaving the predictor architecture and parameter count unchanged.

The Fourier bank B_{F} contains fixed real-valued local discrete Fourier filters. The Random bank replaces the prescribed sinusoidal directions with random directions while preserving every row norm. For each Fourier row b_k^{F}, we sample $u_k \sim \mathcal{N}(0, I_P)$ and set

$$b_k^{\mathrm{R}} = \frac{\|b_k^{\mathrm{F}}\|_2}{\|u_k\|_2} u_k.$$

The two banks share the filter length P, response count K, norm of each corresponding filter, Raw branch, predictor, and training configuration. Fourier uses prescribed sinusoidal directions, while Random uses fixed Gaussian directions. Their performance difference measures the contribution of the prescribed frequency structure. Their shared improvement over Raw measures the contribution of changes common to both expanded inputs. Appendix A.1 specifies the filtering operation, Fourier row order, padding, and readout initialization. Figure 1 illustrates the matched representations on two real histories.

3.2 Polar Decomposition

The matched comparison examines whether prescribed sinusoidal directions outperform random directions. It does not explain why both expanded inputs may improve over Raw. Both banks alter distances between input histories and produce multiple filter responses. Polar decomposition (Higham, 1986; Higham & Schreiber, 1990) separates these effects.

For two histories x and x', their squared distance after filtering is

$$d_B^2(x, x') = \|B(x - x')\|_2^2 = \langle x - x', B^*B(x - x')\rangle .$$

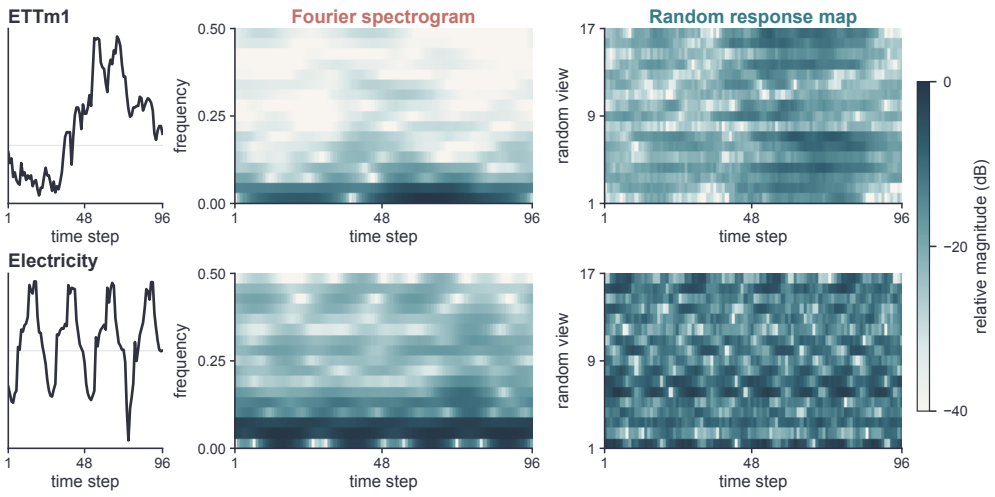


Figure 1: Fourier and Random representations of the same time-series histories. Fourier orders the responses by frequency, whereas Random has no prescribed frequency order. Both expose multiple filtered views of the same input.

The operator B^*B records which temporal differences the bank amplifies or suppresses. Polar decomposition separates the positive semidefinite factor from the partial isometry,

$$B = QH, \qquad H = (B^*B)^{1/2}, \qquad Q = BH^\dagger.$$

Here B^* is the adjoint and $H^\dagger$ is the pseudoinverse. The positive semidefinite factor H maps one length-L sequence to another and satisfies $H^*H = B^*B$, exactly reproducing the metric induced by B. The partial isometry Q maps the resulting sequence to K length-L filter responses while preserving total squared norm over the temporal subspace represented by the bank,

$$\|B\delta\|_2 = \|H\delta\|_2, \qquad \|Qu\|_2 = \|u\|_2 \quad \text{for } u \in (\ker B)^\perp.$$

To create K branches without introducing distinct filter responses, we use the Repeat operator

$$Rx = \left[\frac{x}{\sqrt{K}}, \ldots, \frac{x}{\sqrt{K}}\right], \qquad \|Rx\|_2 = \|x\|_2.$$

The factor $1/\sqrt{K}$ keeps the total squared norm unchanged. Together, Repeat, H, and Q give four matched representations,

$$\begin{aligned} Z_{\text{Repeat}}(x) &= [x, Rx], \\ Z_H(x) &= [x, RHx], \\ Z_Q(x) &= [x, Qx], \\ Z_{QH}(x) &= [x, QHx] = [x, Bx]. \end{aligned}$$

These four conditions isolate repeated branches, the positive semidefinite factor H, distinct norm-preserving filter responses Q, and the full filter bank QH. They retain the same Raw branch and share the predictor, readout width, and training budget. To include the interaction between H and Q, we evaluate each factor both alone and together with the other factor. The effect of H averages the contrasts between the Repeat and H conditions and between Q and QH. The effect of Q averages the contrasts between the Repeat and Q conditions and between H and QH. This averaging decomposes the full gain without assigning the interaction to either factor alone. Appendices A.2 and A.3 give the construction and attribution equations.

Table 1: Evaluation MSE across generic backbones and datasets. Bold and underlined values denote the best and second-best inputs for each backbone and dataset.

Backbone	Input	ETTm1	Weather	Electricity	Exchange
RNN	Raw	.4644 ±.0022	.4866 ±.0009	.1915 ±.0018	**.2702** ±.0106
	Random	**.4480** ±.0063	**.4020** ±.0014	.1901 ±.0008	.2737 ±.0113
	Fourier	.4535 ±.0040	.4250 ±.0318	**.1895** ±.0014	.2713 ±.0117
Transformer	Raw	.4896 ±.0085	.3926 ±.0020	.1595 ±.0007	.3506 ±.0043
	Random	**.4192** ±.0034	**.3917** ±.0027	.1594 ±.0012	**.2075** ±.0352
	Fourier	.4240 ±.0035	.3926 ±.0014	**.1582** ±.0020	.2214 ±.0346
LSTM	Raw	.4883 ±.0095	.3879 ±.0006	.1572 ±.0008	.2181 ±.0148
	Random	**.4400** ±.0136	.3901 ±.0033	.1586 ±.0014	.2159 ±.0144
	Fourier	.4455 ±.0106	**.3862** ±.0025	**.1564** ±.0006	**.1915** ±.0126
GRU	Raw	.4498 ±.0075	**.3867** ±.0015	**.1621** ±.0006	**.2012** ±.0192
	Random	.4265 ±.0037	.3895 ±.0016	.1638 ±.0018	.2177 ±.0204
	Fourier	**.4185** ±.0015	.3871 ±.0010	.1628 ±.0019	.2182 ±.0250
MLP	Raw	.4147 ±.0014	.3908 ±.0011	.1610 ±.0003	.2585 ±.0295
	Random	**.4138** ±.0011	**.3904** ±.0010	.1601 ±.0004	**.2531** ±.0285
	Fourier	.4145 ±.0011	.3905 ±.0008	**.1595** ±.0004	.2569 ±.0298
CNN	Raw	**.4074** ±.0012	.4096 ±.0012	.1635 ±.0003	**.1795** ±.0043
	Random	.4083 ±.0008	.4088 ±.0011	.1632 ±.0003	.1801 ±.0053
	Fourier	.4082 ±.0007	**.4075** ±.0012	**.1632** ±.0002	.1797 ±.0032
GNN	Raw	.5380 ±.0294	.4706 ±.0494	.1897 ±.0025	.4168 ±.0375
	Random	**.5012** ±.0121	**.4143** ±.0032	.1879 ±.0007	.4153 ±.0260
	Fourier	.5164 ±.0312	.4286 ±.0194	**.1868** ±.0008	**.4035** ±.0343
Mamba	Raw	.4972 ±.0089	.3885 ±.0009	.1718 ±.0007	.4404 ±.0173
	Random	**.4667** ±.0061	**.3869** ±.0010	.1694 ±.0007	.3838 ±.0180
	Fourier	.4719 ±.0032	.3871 ±.0005	**.1688** ±.0010	**.3673** ±.0409

3.3 Power-Matched Repeat Control

The partial isometry Q produces K distinct filter responses. To separate response differences from branch count, we compare Q with a power-matched Repeat control R_Q. The control replaces each response with a scaled copy of Raw and matches its average squared norm over the training histories. The two inputs share the same branch count and response power, differing only in whether the additional branches contain distinct responses. We compare Q and R_Q using the main experimental configuration. Appendix A.4 details the construction.

4 Experiments

4.1 Experimental Setup

We evaluate RNN (Elman, 1990), Transformer (Vaswani et al., 2017), LSTM (Hochreiter & Schmidhuber, 1997), GRU (Cho et al., 2014), MLP (Rumelhart et al., 1986), CNN (LeCun et al., 1989), GNN (Scarselli et al., 2009), and Mamba (Gu & Dao, 2024) backbones on ETTm1 (Zhou et al., 2021), Weather[1], Electricity (Trindade, 2015; Lai et al., 2018), and Exchange (Lai et al., 2018). The generic, scaling, and mechanism experiments use a history length and forecast horizon of 96. These histories cover one day on ETTm1, 16 hours on Weather, and four days on Electricity. The primary configuration of each generic backbone contains approximately 0.05 million parameters. Experiments with published models retain the original input configuration of each architecture and use a forecast horizon of 96.

The primary comparison uses $P = 8$, $K = 8$, and an unchanged Raw branch. Within each setting, Raw, Random, and Fourier share the same base-predictor initialization and training protocol. View

[1] https://weather.bgc-jena.mpg.de/weather_data.html

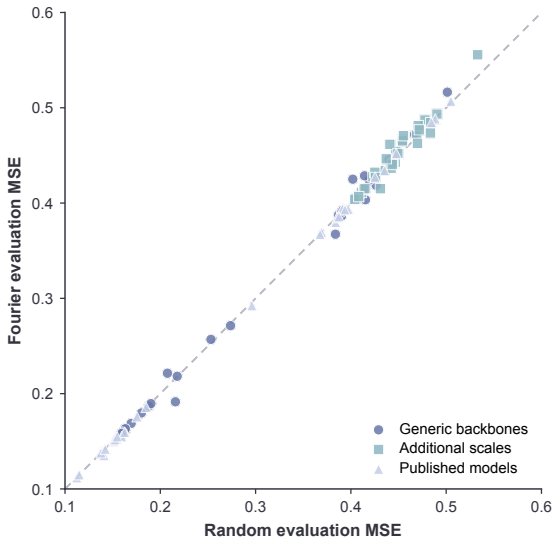


Figure 2: Fourier and Random evaluation MSE across 96 matched settings. Marker shapes distinguish generic backbones, additional scales, and published models. The diagonal marks equal error.

expansion leaves the base predictor unchanged and adds only K response-combination coefficients to the readout. We report evaluation MSE as the mean and sample standard deviation over predictor seeds $0, 1, 2$.

4.2 Aggregate Comparison

The three studies provide 96 matched Fourier and Random comparisons across generic backbones, additional scales, and published models. As shown in Figure 2, the paired results cluster near the diagonal and fall on both sides of it. The overall comparison shows no forecasting advantage for Fourier over Random. The counts for each study appear in Appendix Table 3.

4.3 Evaluation Across Architectures and Scales

4.3.1 Generic Backbones

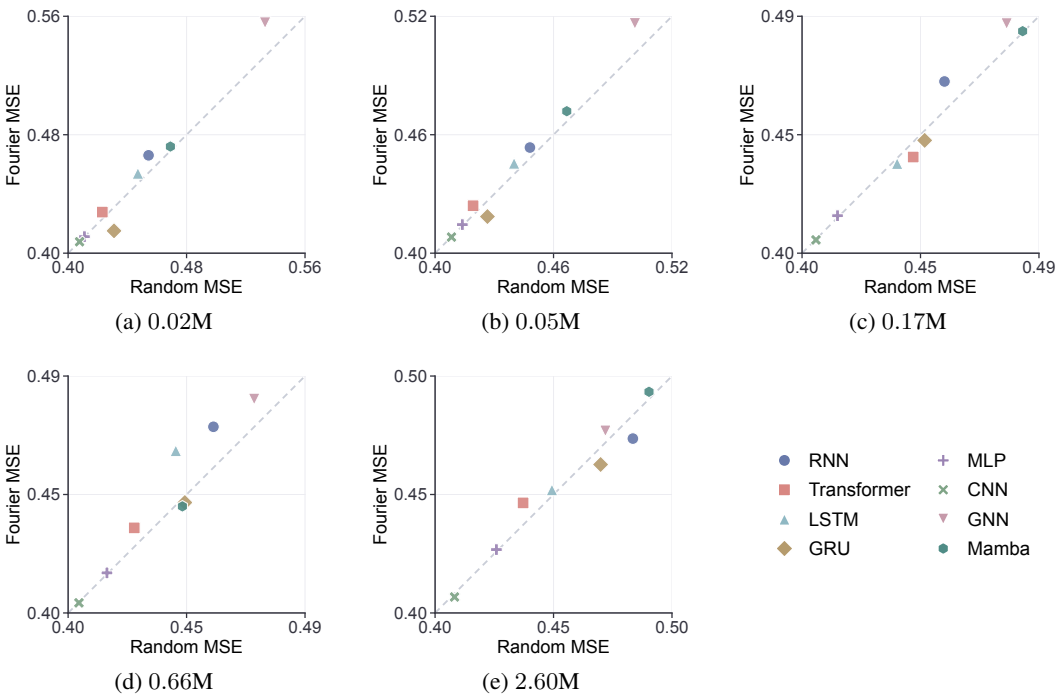
Table 1 reports the complete comparison across generic backbones and datasets. The Raw baseline places the Fourier–Random difference in the context of the improvement from transforming the input. For Transformer on ETTm1 and Exchange, both expanded inputs improve substantially on Raw, while the Fourier–Random difference is much smaller. Their close performance is not confined to settings where all three inputs perform alike. Random retains the gain even when changing the input representation has a clear effect on prediction.

4.3.2 Backbone Scaling

The scaling study examines whether this comparison persists beyond the small backbones used in the primary experiment. We evaluate all eight generic backbones at approximately 0.02, 0.05, 0.17, 0.66, and 2.60 million parameters on ETTm1. At every scale, the paired Fourier and Random errors in Figure 3 lie near and on both sides of the diagonal. The similar performance observed in the primary experiment extends across the evaluated backbone sizes. Complete results appear in Appendix Table 6.

4.3.3 Published Fourier-Based Models

We replace the Fourier filters or bases inside the eight published Fourier-based forecasting models in Table 2 with parameter-matched Random counterparts (Yi et al., 2023b;a; 2024; Eldele et al., 2024; Huang et al., 2025; Liu, 2025; Zhang et al., 2026; An et al., 2026). Parameter counts appear



(a) 0.02M (b) 0.05M (c) 0.17M

(d) 0.66M (e) 2.60M

Figure 3: Fourier and Random evaluation MSE across five ETTm1 backbone scales. Colors and markers identify backbones. The diagonal marks equal error.

in Appendix Table 5. The paired errors are close in most settings, with no substantial forecasting advantage for Fourier.

These substitutions extend the comparison from a shared input interface to architectures developed around Fourier operations. The surrounding learned components are retained and trained with the replacement. Their similar accuracy indicates that these components can make effective use of a fixed representation without the prescribed Fourier basis. The success of the full forecasting architecture need not imply that the replaced basis is essential to its performance.

5 Mechanism Analysis

The experiments in Section 4 show that Fourier and Random perform similarly in aggregate and often improve on Raw. Within the controlled filter-bank framework, we use the decomposition in Section 3.2 to separate input-metric changes, additional response branches, and differences among filter responses. We examine how the shared gain varies across backbones, datasets, response counts, and filter lengths.

5.1 Source of the Shared Gain

Figure 4a presents the decomposition results for both filter families. The positive semidefinite factor H reproduces the input metric induced by the full filter bank, but neither H nor Repeat contributes an aggregate gain. The partial isometry Q accounts for nearly all of the full filter-bank gain over Raw.

The input produced by Q contains K additional branches. Its gain may come either from the additional predictor evaluations and wider readout or from the differences among filter responses. Repeat matches the branch count and total squared norm and replaces every filter response with a scaled copy of Raw. Its lack of aggregate gain shows that additional branches alone are insufficient. In the power-matched Repeat comparisons, distinct filter responses generally yield lower error than repeated inputs with matched branch counts and per-response power, as shown across eight backbones

Table 2: Evaluation MSE for matched Fourier and Random variants of eight published Fourier-based forecasting models. The lower value in each pair is bolded. Exact ties are bolded together.

Model & Pub.	Variant	ETTm1	Weather	Electricity	Exchange
FreTS	Fourier	**.4342** ±.0017	.4520 ±.0006	.1757 ±.0005	**.1548** ±.0107
(NeurIPS 2023)	Random	.4353 ±.0054	**.4476** ±.0032	**.1755** ±.0005	.1595 ±.0087
FourierGNN	Fourier	**.4883** ±.0159	.5069 ±.0178	**.2924** ±.0177	.1857 ±.0118
(NeurIPS 2023)	Random	.4885 ±.0162	**.5050** ±.0174	.2961 ±.0196	**.1851** ±.0139
FilterNet	Fourier	.3937 ±.0024	.3937 ±.0007	**.1350** ±.0008	**.1504** ±.0087
(NeurIPS 2024)	Random	**.3927** ±.0035	**.3915** ±.0043	.1410 ±.0008	.1510 ±.0054
TSLANet	Fourier	.3693 ±.0019	**.4846** ±.0022	**.1118** ±.0003	**.1378** ±.0027
(ICML 2024)	Random	**.3692** ±.0017	**.4846** ±.0022	.1123 ±.0003	**.1378** ±.0027
TimeKAN	Fourier	**.3797** ±.0005	**.3935** ±.0035	.1535 ±.0001	.1533 ±.0012
(ICLR 2025)	Random	.3840 ±.0034	.3971 ±.0011	**.1532** ±.0010	**.1527** ±.0021
FreqMoE	Fourier	.3939 ±.0016	.4273 ±.0010	**.1594** ±.0003	**.1521** ±.0021
(AISTATS 2025)	Random	**.3910** ±.0009	**.4254** ±.0017	.1625 ±.0002	.1539 ±.0017
FreqCycle	Fourier	**.3671** ±.0006	**.3848** ±.0010	.1149 ±.0000	**.1516** ±.0028
(AAAI 2026)	Random	.3677 ±.0005	.3855 ±.0003	**.1149** ±.0001	.1523 ±.0016
FreDN	Fourier	**.3858** ±.0025	**.3930** ±.0031	**.1417** ±.0014	.1548 ±.0023
(AAAI 2026)	Random	.3875 ±.0026	.3938 ±.0032	.1423 ±.0012	**.1547** ±.0041

on ETTm1 in Appendix Table 13. The contribution of Q comes from the differences among its filter responses. Appendix Tables 9 and 12 report the aggregate and backbone-level decompositions.

5.2 Backbone and Dataset Variation

The effect of view expansion varies across forecasting models. Figure 4b pools Random and Fourier over the four datasets for each backbone in the mechanism study. The confidence intervals for Transformer, Mamba, GNN, and RNN lie entirely above zero. LSTM has a positive mean gain with an interval that crosses zero. MLP and CNN remain close to Raw. The direction of the GRU effect varies across datasets.

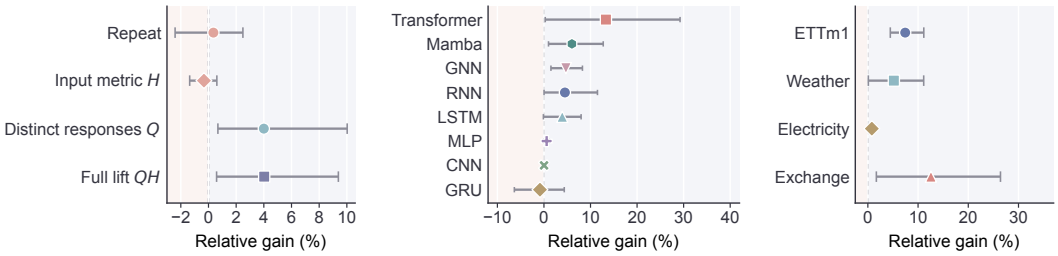
Figure 4c compares view-expansion gains across datasets for GNN, LSTM, Mamba, RNN, and Transformer. These five backbones have the largest mean gains in Figure 4b. The expanded inputs provide the largest gains on ETTm1 and Exchange, while their performance on Electricity remains close to Raw. Appendix Tables 10 and 11 report the corresponding numerical results.

However, the lack of an overall advantage does not mean that Fourier representations are without value. At the 96-point history length, ETTm1 and Weather span one day and 16 hours, respectively. Their input windows do not contain multiple repetitions of the daily cycle. Electricity histories span four days and exhibit strong repeated daily patterns. On this dataset, Fourier yields lower mean MSE than Random across all eight generic backbones (Table 1). These results suggest that prescribed frequency structure can provide additional predictive value when it matches repeated patterns in the observed history. Its potential benefit comes from this relationship to the data, rather than from the human interpretability of a Fourier basis alone.

5.3 Response Count and Filter Length

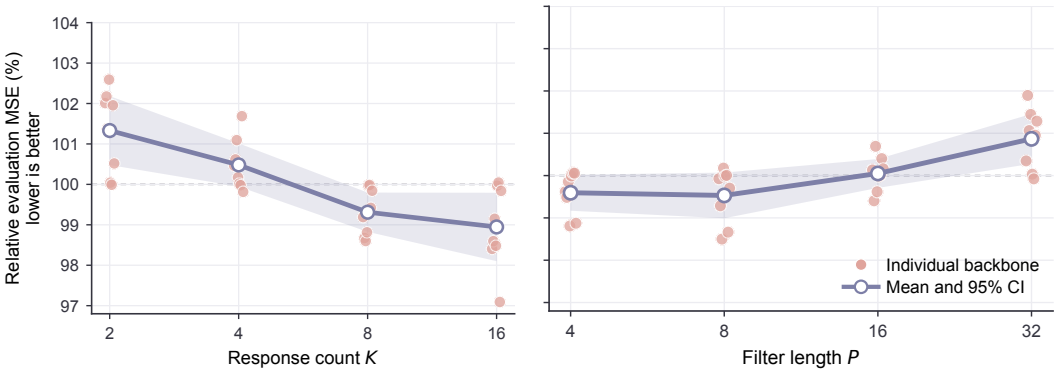
We fix the original filter length at $P = 8$ and vary the response count $K \in \{2, 4, 8, 16\}$ in the partial-isometry component Q. At each K, we rebuild the filter bank and recompute Q. Figure 5a pools Random and Fourier across all eight backbones on ETTm1. Evaluation MSE generally decreases as the response count grows. The magnitude of the change varies across backbones.

We fix $K = 8$ and vary the original filter length $P \in \{4, 8, 16, 32\}$ for the same Q components, backbones, and dataset. Changing P alters the original filters and the resulting Q responses while keeping the branch count fixed. Figure 5b shows similar performance at $P = 4$ and $P = 8$, with no improvement at $P = 16$ or $P = 32$. The complete backbone-level results appear in Appendix



(a) Source of shared gain (b) Backbone variation (c) Dataset variation

Figure 4: Attribution and scope of the gain shared by Random and Fourier, expressed as relative MSE reductions with 95% confidence intervals. Panel (a) decomposes the gain into Repeat, H, and Q. Panels (b) and (c) summarize the gain by backbone and dataset.



(a) Response count at fixed $P = 8$ (b) Filter length at fixed $K = 8$

Figure 5: Response-count and filter-length ablations on ETTm1, pooling Random and Fourier across eight backbones. MSE is normalized within each backbone, filter family, and predictor seed by its geometric mean over the sweep.

Tables 7 and 8. Under this filter-bank construction, increasing the response count generally improves forecasting, whereas extending the original filters provides no corresponding benefit.

6 Conclusion

Fourier-based time-series forecasters are commonly interpreted as benefiting from the periodic structure represented by a prescribed Fourier basis. Our matched comparisons separate the prescribed frequency structure from the accompanying view expansion. Across 96 settings spanning eight generic backbones, five backbone scales, four datasets, and eight published models, Fourier provides no aggregate forecasting advantage over matched Random representations.

Our analysis attributes the gain shared by Fourier and Random primarily to multiple distinct filtered views of the same history. Changes to the input metric and repeated copies of Raw provide no aggregate improvement. Increasing the response count usually lowers forecasting error, while increasing filter length provides no corresponding benefit. Prescribed frequency structure can provide additional gains when it matches repeated patterns within the input window. These results motivate future research on learned or data-adaptive bases that produce compact sets of distinct filter responses. Such bases should match the temporal structure within the input window, rather than prioritize frequency structures that are easy for humans to interpret.

AI USE STATEMENT

Generative AI tools assisted with code development and language editing. The authors verified all AI-assisted content and take full responsibility for the final manuscript.

REFERENCES

Zhongde An, Jinhong You, Jiyanglin Li, Yiming Tang, Wen Li, Heming Du, and Shouguo Du. FreDN: Spectral disentanglement for time series forecasting via learnable frequency decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19623–19631, 2026. doi: 10.1609/aaai.v40i24.39042.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014. doi: 10.3115/v1/D14-1179.

Elisha Dayag, Nhat Thanh Tran, and Jack Xin. Filter then attend: Improving attention-based time series forecasting with spectral filtering. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1131–1135, 2026. doi: 10.1109/ICASSP55912.2026.11462613.

Berken Utku Demirel and Christian Holz. Learning without augmenting: Unsupervised time series representation learning via frame projections. In *Advances in Neural Information Processing Systems*, volume 38, pp. 121916–121961, 2025. doi: 10.52202/085713-4064.

Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020. doi: 10.1007/s10618-020-00701-z.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, and Xiaoli Li. TSLANet: Rethinking Transformers for time series representation learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12409–12428, 2024.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1.

Amir-massoud Farahmand, Sepideh Pourazarm, and Daniel Nikovski. Random projection filter bank for time series data. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6562–6572, 2017.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986. doi: 10.1137/0907079.

Nicholas J. Higham and Robert S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM Journal on Scientific and Statistical Computing*, 11(4):648–655, 1990. doi: 10.1137/0911038.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *International Conference on Learning Representations*, pp. 28514–28529, 2025.

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. doi: 10.1145/3209978.3210006.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4296–4313, 2022. doi: 10.18653/v1/2022.naacl-main.319.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021. doi: 10.1098/rsta.2020.0209.

Ziqi Liu. FreqMoE: Enhancing time series forecasting through frequency decomposition mixture of experts. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3430–3438, 2025.

Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiyao Lin. BasisFormer: Attention-based time series forecasting with learnable and interpretable basis. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71222–71241, 2023. doi: 10.52202/075280-3118.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency debiased Transformer for time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2400–2410, 2024. doi: 10.1145/3637528.3671928.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1177–1184, 2007.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020.

William Toner and Luke Nicholas Darlow. An analysis of linear time series forecasting models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48404–48427, 2024.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. URL `https://doi.org/10.24432/C58C86`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Hao Wang, Licheng Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. FreDF: Learning to forecast in the frequency domain. In *International Conference on Learning Representations*, pp. 6893–6922, 2025a.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, pp. 16980–17016, 2025b.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *International Conference on Learning Representations*, pp. 26295–26318, 2024.

Runze Yang, Longbing Cao, Jianxun Li, and Jie Yang. Rethinking Fourier transform from a basis functions perspective for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8515–8540, 2024. doi: 10.52202/079017-0272.

Chin-Chia Michael Yeh, Yujie Fan, Xin Dai, Uday Singh Saini, Vivian Lai, Prince Osei Aboagye, Junpeng Wang, Huiyuan Chen, Yan Zheng, Zhongfang Zhuang, Liang Wang, and Wei Zhang. RPMixer: Shaking up time series forecasting with random projections for large spatial-temporal data. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3919–3930, 2024. doi: 10.1145/3637528.3671881.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69638–69660, 2023a. doi: 10.52202/075280-3050.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Defu Lian, Ning An, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76656–76679, 2023b. doi: 10.52202/075280-3349.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. FilterNet: Harnessing frequency filters for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 55115–55140, 2024. doi: 10.52202/079017-1749.

Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. OLinear: A linear model for time series forecasting in orthogonally transformed domain. In *Advances in Neural Information Processing Systems*, volume 38, pp. 60573–60632, 2025. doi: 10.52202/085713-2025.

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. FreqCycle: A multi-scale time-frequency analysis method for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):28159–28167, 2026. doi: 10.1609/aaai.v40i33.40042.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed Transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286, 2022.

A Supplementary Material

This appendix provides the implementation and mathematical details of the matched inputs and polar decomposition. It also reports the complete scaling, response-count, filter-length, and mechanism results.

A.1 Expanded Inputs

A.1.1 Tensor Layout

The input history is $X \in \mathbb{R}^{L \times C}$, where L is the number of time points and C is the number of variables. Every filter is applied independently to each variable. A filter bank contains K rows $b_k \in \mathbb{R}^P$, where P is the filter length and K is the response count.

For a single-variable history $x \in \mathbb{R}^L$, the k-th filter response is

$$(Bx)_{t,k} = \sum_{p=0}^{P-1} b_{k,p} x_{(t+p-\lfloor P/2 \rfloor) \bmod L}.$$

Here $t \in \{0, \ldots, L-1\}$ and $k \in \{1, \ldots, K\}$. This definition uses unit stride and circular padding with $\lfloor P/2 \rfloor$ samples on the left and $P-1-\lfloor P/2 \rfloor$ samples on the right. The output length therefore remains L. Applying the bank to every variable produces a tensor in $\mathbb{R}^{L \times C \times K}$. We prepend the unchanged input along the last axis, which produces $Z_B(X) \in \mathbb{R}^{L \times C \times (K+1)}$. Index zero is always the exact Raw branch.

A.1.2 Filter Banks

The Fourier bank represents a local discrete Fourier transform with separate real-valued cosine and sine filters. Let

$$M = \max(P, K), \qquad s_p = M^{-1/2} w_p, \qquad p \in \{0, \ldots, P-1\},$$

where the experiments use the nonperiodic Hann window

$$w_p = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi p}{P-1}\right)\right].$$

The ordered list of real Fourier rows begins with the constant row s_p. For each positive frequency $m \in \{1, \ldots, \lfloor (M-1)/2 \rfloor\}$, it then contains

$$b^{\mathrm{F},c}_{m,p} = s_p \cos\left(\frac{2\pi m p}{M}\right), \qquad b^{\mathrm{F},s}_{m,p} = -s_p \sin\left(\frac{2\pi m p}{M}\right).$$

When M is even, the final available row is the real Nyquist row $s_p \cos(\pi p)$. The bank retains the first K rows in this order. This construction makes P and K independent.

The Random bank removes the sinusoidal directions while retaining the norm of every corresponding Fourier row. It draws $u_k \sim \mathcal{N}(0, I_P)$, where I_P is the P-dimensional identity matrix, and forms

$$b^{\mathrm{R}}_k = \frac{\|b^{\mathrm{F}}_k\|_2}{\|u_k\|_2} u_k.$$

Random and Fourier banks contain no trainable parameters.

A.1.3 Shared Predictor

Let F_θ denote the base predictor instantiated by a forecasting backbone. Its parameters are shared across response branches. The predictor maps the Raw branch and every filter response from the same history to a forecast. A linear readout with coefficients $a_0, \ldots, a_K$ combines the branch-wise forecasts

$$\widehat{Y} = \sum_{j=0}^{K} a_j F_\theta(Z_B(X)_j) + \beta.$$

Here $Z_B(X)_j$ is the j-th branch, $j = 0$ is Raw, and β is the readout bias. The readout is shared across forecast positions and variables. It is initialized with $a_0 = 1$, $a_j = 0$ for $j \geq 1$, and $\beta = 0$. Every expanded model therefore begins with the same predictions as the corresponding Raw model. The expanded input changes the number of predictor evaluations and adds K readout weights, but it does not alter the base predictor or its parameter count.

A.2 Polar Decomposition

A.2.1 Frequency Construction

Circular filtering is diagonal in the discrete Fourier basis. Let $\widehat{x}(\omega)$ be the discrete Fourier transform of x, and let $\widehat{b}_k(\omega)$ be the transfer function of the k-th filter under the indexing defined in Appendix A.1. We define

$$h(\omega) = \left(\sum_{k=1}^{K} \left| \widehat{b}_k(\omega) \right|^2 \right)^{1/2}, \qquad q_k(\omega) = \begin{cases} \widehat{b}_k(\omega)/h(\omega), & h(\omega) > 0, \\ 0, & h(\omega) = 0. \end{cases}$$

The same-dimensional positive semidefinite factor H and the partial isometry Q act as

$$\widehat{Hx}(\omega) = h(\omega)\widehat{x}(\omega), \qquad \widehat{Q_k x}(\omega) = q_k(\omega)\widehat{x}(\omega).$$

The scalar $h(\omega)$ records the combined gain of all filters at frequency ω. The vector $(q_1(\omega), \ldots, q_K(\omega))$ records how that gain is distributed across filter responses. On every represented frequency,

$$\sum_{k=1}^{K} |q_k(\omega)|^2 = 1, \qquad \widehat{Q_k H x}(\omega) = \widehat{b}_k(\omega)\widehat{x}(\omega).$$

Consequently, Q preserves total squared norm on the represented subspace and $QH = B$. The Raw branch is not factorized and remains unchanged in every condition.

A.3 Component Attribution

Let $\mathcal{E}(S)$ be the mean evaluation MSE over predictor seeds $0, 1, 2$ with input S, where lower values are better. We use H, Q, and QH as shorthand for the inputs Z_H, Z_Q, and Z_{QH} defined in Section 3.2. We compute every contrast within the same backbone, dataset, filter family, and training budget. The effect of adding repeated inputs is

$$\Delta_{\text{rep}} = \mathcal{E}(\text{Raw}) - \mathcal{E}(\text{Repeat}).$$

The contributions of H and Q are the averages of their marginal improvements across the two possible insertion orders

$$\phi_H = \frac{1}{2} \left[\mathcal{E}(\text{Repeat}) - \mathcal{E}(H) + \mathcal{E}(Q) - \mathcal{E}(QH) \right],$$

$$\phi_Q = \frac{1}{2} \left[\mathcal{E}(\text{Repeat}) - \mathcal{E}(Q) + \mathcal{E}(H) - \mathcal{E}(QH) \right].$$

Positive values indicate improvement. The three components exactly recover the full filter-bank gain

$$\mathcal{E}(\text{Raw}) - \mathcal{E}(QH) = \Delta_{\text{rep}} + \phi_H + \phi_Q.$$

The Repeat term captures the combined contribution of presenting K additional copies, evaluating the predictor on them, and fitting the wider readout. The term ϕ_H measures the average contribution of the positive semidefinite factor. The term ϕ_Q measures the average contribution of replacing repeated inputs with norm-preserving filter responses.

Tables 9–12 use relative MSE reduction $100[\mathcal{E}(\text{Raw}) - \mathcal{E}(S)]/\mathcal{E}(\text{Raw})$. Each attribution term is divided by the same Raw mean and multiplied by 100 before averaging equally over the included backbones, datasets, and filter families. The 95% confidence intervals use 100,000 percentile bootstrap resamples of dataset blocks for each backbone, backbone blocks for each dataset, and both

Table 3: Counts of Fourier wins, Random wins, and ties across the generic, scaling, and published-model studies. The scaling group excludes the 0.05M configurations already included in the generic group.

Study	Paired settings	Fourier wins	Random wins	Ties
Generic backbones	32	18	14	0
Additional scales	32	12	20	0
Published models	32	17	13	2
All	96	47	47	2

axes independently for overall attribution. Both filter families and all input conditions remain paired within each block. These intervals summarize variation across backbones and datasets.

A.4 Power-Matched Repeat Control

A.4.1 Power Matching

The operator Q can distribute different amounts of empirical power to its filter responses. We construct a separate power-matched Repeat control for each filter family. For training histories $X \sim \mathcal{D}_{\text{train}}$, the scale of response k is

$$c_k^2 = \frac{\mathbb{E}\|Q_k X\|_2^2}{\mathbb{E}\|X\|_2^2}, \qquad R_Q X = [c_1 X, \ldots, c_K X].$$

Each response of $R_Q X$ matches the average squared norm of the corresponding response of QX, while all K responses remain collinear copies of Raw. Table 13 compares Q and R_Q across all eight backbones on ETTm1 at $P = 8$ and $K = 8$.

A.4.2 Response Count

The response-count experiment fixes $P = 8$ and evaluates $K \in \{2, 4, 8, 16\}$ for all eight backbones on ETTm1. Random and Fourier filter banks are rebuilt at every K.

The Fourier construction uses $M = \max(P, K)$. Values $K \leq 8$ select the first K rows from the real-valued P-point Fourier filter list defined in Appendix A.1.2. The setting $K = 16$ uses a 16-point frequency grid applied over the same filter length $P = 8$. Comparisons across different K measure how forecasting changes with the response count under this construction.

The complete eight-backbone K results appear in Table 7. The filter-length control in Table 8 fixes $K = 8$ and evaluates $P \in \{4, 8, 16, 32\}$ on ETTm1 with the same eight backbones.

A.5 Experimental Details

A.5.1 Datasets and Backbones

The generic, scaling, and mechanism experiments use a history length of 96 and a forecast horizon of 96. Experiments with published models retain the original history length of each architecture and use a forecast horizon of 96. The generic training implementation uses at most 32 variables in each optimization batch and an effective batch size of 32.

The MLP applies a dense projection over the full history, the CNN uses a temporal convolutional architecture, and the GNN performs variable-wise graph message passing before a recurrent sequence encoder. Within each backbone and dataset, Raw, Random, and Fourier use the same base-predictor initialization, optimizer, data order, batch size, schedule, and training budget.

TSLANet uses a history length and patch size of 64 on Exchange and Weather, leaving a single input patch. Its Fourier and Random orthogonal transforms are therefore identical, producing the two exact ties in Table 3.

Table 4: Base predictor parameter counts and learning rates used in the primary experiments.

Backbone	Base predictor parameters	Learning rate
RNN	48,496	3×10^{-4}
Transformer	49,480	5×10^{-3}
LSTM	47,328	1×10^{-2}
GRU	49,488	3×10^{-3}
MLP	50,289	3×10^{-4}
CNN	50,083	3×10^{-4}
GNN	49,000	3×10^{-4}
Mamba	49,311	1×10^{-3}

Table 5: Parameter counts in millions for the published models under each dataset configuration.

Model	ETTm1	Weather	Electricity	Exchange
FreTS	3.2368	3.2368	3.2368	3.2368
FourierGNN	0.2069	0.2069	0.2069	0.2069
FilterNet	0.0496	0.0718	0.6807	0.0496
TSLANet	0.9058	0.1976	0.8806	0.1976
TimeKAN	0.0144	0.0209	0.0233	0.0144
FreqMoE	0.0457	0.0457	0.0457	0.0457
FreqCycle	0.0511	0.0533	0.1044	0.0505
FreDN	0.1151	0.0818	0.3698	0.1155

The eight backbones are matched to an approximately 49,000-parameter base predictor budget. Table 4 gives the exact base predictor counts and learning rates. The expanded input adds only the response-combination coefficients described in Appendix A.1.3. For a small number of settings with unstable evaluation MSE or high cross-seed variance, we adjusted the learning rate to obtain stable convergence. Raw, Random, and Fourier use the same learning rate within every reported setting.

The Mamba backbone uses the official `mamba_ssm.Mamba` implementation with state dimension 16, expansion factor 2, and causal convolution width 4. The primary configuration has width 51 and two Mamba blocks. The scaling configurations use widths 42, 51, 84, 152, and 253 with 1, 2, 3, 4, and 6 blocks, giving 20,004, 49,311, 169,440, 661,448, and 2,595,117 predictor parameters. Learning rates are selected once on the ETTm1 training prefix for each scale and then shared by Raw, Random, and Fourier. The selected rates are 3×10^{-3}, 1×10^{-3}, 1×10^{-3}, 3×10^{-3}, and 1×10^{-4} in increasing scale order.

The generic, scaling, and mechanism models use AdamW with weight decay 10^{-4} and a maximum budget of 10,000 training steps. We evaluate the models every 100 steps. Experiments with published models use a maximum budget of 3,000 steps. Predictor seeds are exactly $0, 1, 2$.

A.5.2 Parameter Counts

Table 5 reports parameter counts for the eight published models under each dataset configuration. Fourier and Random variants have the same parameter counts.

A.6 Complete Scaling Results

Complete scaling results are reported in Table 6.

A.7 Complete Response-Count and Filter-Length Results

Complete response-count and filter-length results are reported in Tables 7 and 8.

Table 6: Evaluation MSE across ETTm1 backbone scales. Bold and underlined values denote the best and second-best inputs for each backbone and scale.

Backbone	Input	≈0.02M	≈0.05M	≈0.17M	≈0.66M	≈2.60M
RNN	Raw	.4806 ±.0044	.4644 ±.0022	.4848 ±.0037	.4927 ±.0079	.5125 ±.0131
	Random	**.4543** ±.0055	**.4480** ±.0063	**.4541** ±.0077	**.4552** ±.0009	.4835 ±.0105
	Fourier	.4661 ±.0004	.4535 ±.0040	.4652 ±.0057	.4707 ±.0026	**.4736** ±.0100
Transformer	Raw	.4891 ±.0064	.4896 ±.0085	.5071 ±.0116	.4926 ±.0108	.4923 ±.0155
	Random	**.4231** ±.0134	**.4192** ±.0034	.4422 ±.0109	**.4251** ±.0064	**.4371** ±.0057
	Fourier	.4278 ±.0093	.4240 ±.0035	**.4365** ±.0111	.4323 ±.0060	.4465 ±.0070
LSTM	Raw	.4831 ±.0123	.4883 ±.0095	.4796 ±.0084	.4850 ±.0135	.4969 ±.0188
	Random	**.4471** ±.0060	**.4400** ±.0136	.4361 ±.0048	**.4409** ±.0026	**.4493** ±.0058
	Fourier	.4539 ±.0167	.4455 ±.0106	**.4340** ±.0007	.4616 ±.0064	.4520 ±.0157
GRU	Raw	.4496 ±.0076	.4498 ±.0075	.4665 ±.0087	.4799 ±.0066	.4852 ±.0127
	Random	.4310 ±.0046	.4265 ±.0037	.4466 ±.0115	.4443 ±.0070	.4699 ±.0097
	Fourier	**.4151** ±.0044	**.4185** ±.0015	**.4429** ±.0066	**.4419** ±.0075	**.4627** ±.0086
MLP	Raw	.4120 ±.0011	.4147 ±.0014	.4142 ±.0037	.4153 ±.0023	.4271 ±.0028
	Random	**.4110** ±.0010	**.4138** ±.0011	**.4135** ±.0033	**.4148** ±.0017	**.4259** ±.0003
	Fourier	.4112 ±.0008	.4145 ±.0011	.4142 ±.0033	.4152 ±.0018	.4268 ±.0002
CNN	Raw	**.4067** ±.0012	**.4074** ±.0012	**.4049** ±.0012	**.4033** ±.0011	**.4057** ±.0015
	Random	.4077 ±.0015	.4083 ±.0008	.4053 ±.0013	.4041 ±.0024	.4082 ±.0032
	Fourier	.4077 ±.0012	.4082 ±.0007	.4050 ±.0014	.4038 ±.0025	.4068 ±.0032
GNN	Raw	.5682 ±.0101	.5380 ±.0294	.5074 ±.0075	.5308 ±.0208	.4826 ±.0106
	Random	**.5330** ±.0148	**.5012** ±.0121	**.4777** ±.0031	**.4707** ±.0088	**.4719** ±.0072
	Fourier	.5557 ±.0286	.5164 ±.0312	.4872 ±.0101	.4813 ±.0179	.4768 ±.0043
Mamba	Raw	.5120 ±.0046	.4972 ±.0089	.5099 ±.0057	.4795 ±.0054	.5004 ±.0031
	Random	**.4691** ±.0089	**.4667** ±.0061	**.4838** ±.0148	.4434 ±.0098	**.4903** ±.0041
	Fourier	.4720 ±.0032	.4719 ±.0032	.4843 ±.0140	**.4405** ±.0023	.4934 ±.0043

A.8 Aggregate Mechanism and Scope Results

Aggregate mechanism, backbone, and dataset results are reported in Tables 9–11.

A.9 Detailed Mechanism Results

Per-backbone mechanism results are reported in Table 12.

Architectural interpretation. Fourier and Random remain comparable across backbones, while their shared gain over Raw depends on how the backbone mixes temporal information. An MLP begins with a dense projection over the complete history, so fixed filter responses can be absorbed into its learned temporal projection. A CNN composes each fixed input filter with learned convolutional kernels, which makes the added responses largely redundant. This representational overlap agrees with the near-zero aggregate gains of MLP and CNN.

Transformer, Mamba, and GNN process each filter response through attention, selective state updates, or graph and recurrent interactions before combining forecasts. Distinct responses therefore provide several computational paths from the same history. These backbones show the clearest gains, with their improvements attributed mainly to Q rather than H in Table 12. RNN and LSTM form an intermediate group. Their recurrent states can learn temporal mixtures but compress the history sequentially, which may explain why their gains vary more across datasets. GRU is the least stable recurrent case. Its positive Q contribution is offset by the negative Repeat contribution, which may reflect a less stable interaction between additional response branches and gated state updates.

Table 7: Evaluation MSE across response counts K for the partial-isometry component Q on ETTm1 at $P = 8$. Within each filter family, bold and underlined values denote the best and second-best settings.

Backbone	Filter family	$K = 2$	$K = 4$	$K = 8$	$K = 16$
RNN	Random	0.4595 ± 0.0060	0.4465 ± 0.0063	<u>0.4365</u> ± 0.0093	**0.4326** ± 0.0045
	Fourier	0.4511 ± 0.0047	0.4504 ± 0.0114	<u>0.4490</u> ± 0.0030	**0.4459** ± 0.0019
Transformer	Random	0.4364 ± 0.0011	0.4238 ± 0.0040	<u>0.4175</u> ± 0.0014	**0.4140** ± 0.0044
	Fourier	0.4243 ± 0.0043	0.4238 ± 0.0074	**0.4137** ± 0.0025	<u>0.4166</u> ± 0.0096
LSTM	Random	0.4469 ± 0.0212	0.4489 ± 0.0195	**0.4187** ± 0.0136	<u>0.4343</u> ± 0.0082
	Fourier	0.4455 ± 0.0178	<u>0.4409</u> ± 0.0135	0.4493 ± 0.0035	**0.4385** ± 0.0068
GRU	Random	0.4408 ± 0.0029	0.4298 ± 0.0061	<u>0.4229</u> ± 0.0041	**0.4189** ± 0.0043
	Fourier	0.4325 ± 0.0029	0.4229 ± 0.0031	**0.4183** ± 0.0026	<u>0.4194</u> ± 0.0064
MLP	Random	0.4142 ± 0.0013	0.4141 ± 0.0013	**0.4138** ± 0.0012	<u>0.4139</u> ± 0.0011
	Fourier	0.4145 ± 0.0012	0.4145 ± 0.0012	<u>0.4144</u> ± 0.0011	**0.4142** ± 0.0011
CNN	Random	**0.4079** ± 0.0009	0.4081 ± 0.0007	<u>0.4079</u> ± 0.0008	0.4081 ± 0.0007
	Fourier	0.4079 ± 0.0013	**0.4077** ± 0.0011	<u>0.4079</u> ± 0.0008	0.4081 ± 0.0007
GNN	Random	0.5199 ± 0.0142	0.5119 ± 0.0170	<u>0.4890</u> ± 0.0103	**0.4810** ± 0.0099
	Fourier	<u>0.5039</u> ± 0.0226	0.5091 ± 0.0120	0.5095 ± 0.0273	**0.4937** ± 0.0068
Mamba	Random	0.4766 ± 0.0040	<u>0.4688</u> ± 0.0025	**0.4671** ± 0.0054	0.4697 ± 0.0013
	Fourier	**0.4684** ± 0.0026	0.4697 ± 0.0038	0.4717 ± 0.0050	<u>0.4691</u> ± 0.0066

Table 8: Evaluation MSE across filter lengths P for the partial-isometry component Q on ETTm1 at $K = 8$. Within each filter family, bold and underlined values denote the best and second-best settings.

Backbone	Filter family	$P = 4$	$P = 8$	$P = 16$	$P = 32$
RNN	Random	0.4397 ± 0.0041	<u>0.4365</u> ± 0.0093	**0.4362** ± 0.0034	0.4398 ± 0.0043
	Fourier	**0.4430** ± 0.0087	<u>0.4490</u> ± 0.0030	0.4512 ± 0.0068	0.4494 ± 0.0032
Transformer	Random	<u>0.4189</u> ± 0.0001	**0.4175** ± 0.0014	0.4216 ± 0.0070	0.4428 ± 0.0026
	Fourier	0.4140 ± 0.0020	0.4137 ± 0.0025	**0.4107** ± 0.0077	<u>0.4108</u> ± 0.0094
LSTM	Random	<u>0.4352</u> ± 0.0006	**0.4187** ± 0.0136	0.4374 ± 0.0163	0.4475 ± 0.0039
	Fourier	<u>0.4443</u> ± 0.0050	0.4493 ± 0.0035	0.4497 ± 0.0132	**0.4425** ± 0.0086
GRU	Random	**0.4161** ± 0.0050	0.4229 ± 0.0041	<u>0.4162</u> ± 0.0057	0.4329 ± 0.0097
	Fourier	**0.4135** ± 0.0080	<u>0.4183</u> ± 0.0026	0.4202 ± 0.0024	0.4189 ± 0.0014
MLP	Random	0.4141 ± 0.0012	<u>0.4138</u> ± 0.0012	**0.4137** ± 0.0012	0.4141 ± 0.0012
	Fourier	**0.4141** ± 0.0012	<u>0.4144</u> ± 0.0011	0.4144 ± 0.0012	0.4144 ± 0.0012
CNN	Random	0.4081 ± 0.0008	<u>0.4079</u> ± 0.0008	0.4083 ± 0.0006	**0.4076** ± 0.0014
	Fourier	0.4081 ± 0.0008	0.4079 ± 0.0008	<u>0.4076</u> ± 0.0010	**0.4075** ± 0.0018
GNN	Random	0.5075 ± 0.0192	**0.4890** ± 0.0103	<u>0.4973</u> ± 0.0064	0.5079 ± 0.0091
	Fourier	**0.5049** ± 0.0231	<u>0.5095</u> ± 0.0273	0.5189 ± 0.0299	0.5132 ± 0.0075
Mamba	Random	**0.4640** ± 0.0031	<u>0.4671</u> ± 0.0054	0.4679 ± 0.0050	0.4762 ± 0.0050
	Fourier	**0.4669** ± 0.0040	<u>0.4717</u> ± 0.0050	0.4752 ± 0.0009	0.4774 ± 0.0017

A.10 Power-Matched Repeat Results

Table 13 reports the direct comparison between Q and the power-matched Repeat control R_Q defined in Appendix A.4.1. At matched branch counts and per-response power, distinct filter responses lower evaluation MSE for RNN, Transformer, LSTM, GRU, GNN, and Mamba in both filter families. MLP and CNN show little change. These results show that the benefit of distinct responses persists after matching the power of each branch.

Table 9: Attribution of the gain shared by Random and Fourier. The H and Q rows average marginal contributions over both insertion orders. Positive values denote relative evaluation MSE reductions.

Input change	Attributed gain	95% CI
Repeat	+0.3550%	[−2.4131, 2.4876]
Positive semidefinite factor H	−0.3288%	[−1.3595, 0.6048]
Partial isometry Q	+4.0001%	[0.6805, 10.0236]
Full QH	+4.0263%	[0.5853, 9.3868]

Table 10: Backbone response to view expansion in the mechanism study, pooling Random and Fourier over four datasets. Positive values denote relative evaluation MSE reductions.

Backbone	Gain over Raw	95% CI
Transformer	+13.3144%	[0.2781, 29.2320]
Mamba	+6.0051%	[0.9784, 12.7325]
GNN	+4.7162%	[1.5002, 8.2676]
RNN	+4.5024%	[0.0211, 11.4935]
LSTM	+3.9299%	[−0.1203, 7.9802]
MLP	+0.5795%	[0.1143, 1.0450]
CNN	+0.0211%	[−0.2204, 0.2626]
GRU	−0.8586%	[−6.3581, 4.3632]

Table 11: Dataset response to view expansion in the mechanism study over GNN, LSTM, Mamba, RNN, and Transformer. Positive values denote relative evaluation MSE reductions.

Dataset	Gain over Raw	95% CI
ETTm1	+7.4358%	[4.4713, 11.1419]
Weather	+5.1785%	[0.0967, 11.1225]
Electricity	+0.7939%	[0.2249, 1.3016]
Exchange	+12.5661%	[1.6929, 26.4141]

Table 12: Per-backbone mean contributions to the full filter-bank gain, expressed as percentages of Raw MSE and averaged over four datasets and both filter families. Repeat, the positive semidefinite factor H, and the partial isometry Q sum to the full QH gain. Bold marks the largest positive component in each row.

Backbone	Repeat	H	Q	Full QH
RNN	**+4.3079**	-0.1329	+0.3273	+4.5024
Transformer	+2.2591	-2.6111	**+13.6665**	+13.3144
LSTM	+0.7747	+1.1205	**+2.0348**	+3.9299
GRU	-5.0972	+0.1321	**+4.1065**	-0.8586
MLP	-0.0697	+0.0920	**+0.5573**	+0.5795
CNN	**+0.2801**	-0.0568	-0.2022	+0.0211
GNN	+1.5563	-0.5037	**+3.6635**	+4.7162
Mamba	-1.1709	-0.6708	**+7.8468**	+6.0051

Table 13: Direct comparison of distinct filter responses Q and power-matched repeated inputs R_Q on ETTm1 at $P = 8$ and $K = 8$. Entries are mean evaluation MSE with standard deviations. Bold marks the lower mean within each filter family.

	Random		Fourier	
Backbone	R_Q	Q	R_Q	Q
RNN	0.4488 ±0.0058	**0.4365** ±0.0093	0.4588 ±0.0104	**0.4490** ±0.0030
Transformer	0.4551 ±0.0092	**0.4175** ±0.0014	0.4453 ±0.0044	**0.4137** ±0.0025
LSTM	0.4727 ±0.0029	**0.4187** ±0.0136	0.4692 ±0.0043	**0.4493** ±0.0035
GRU	0.4483 ±0.0023	**0.4229** ±0.0041	0.4320 ±0.0024	**0.4183** ±0.0026
MLP	0.4140 ±0.0012	**0.4138** ±0.0012	0.4148 ±0.0013	**0.4144** ±0.0011
CNN	**0.4070** ±0.0009	0.4079 ±0.0008	**0.4072** ±0.0009	0.4079 ±0.0008
GNN	0.5105 ±0.0273	**0.4890** ±0.0103	0.5206 ±0.0211	**0.5095** ±0.0273
Mamba	0.4984 ±0.0135	**0.4671** ±0.0054	0.4983 ±0.0070	**0.4717** ±0.0050