AUTOCORRELATION REINTRODUCES SPECTRAL BIAS IN KANS FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing theory suggests that Kolmogorov-Arnold Networks (KANs) can overcome the spectral bias commonly observed in neural networks under the assumption that inputs are statistically independent. However, this assumption does not hold in time series forecasting (TSF), where inputs are lagged observations with strong temporal autocorrelation. Through theoretical analysis and empirical validation, we obtain an unexpected finding: temporal autocorrelation reintroduces spectral bias in KANs, and the bias becomes increasingly pronounced as the degree of autocorrelation increases. This suggests that standard KANs may face substantial difficulties in TSF with strongly autocorrelated inputs. To address this problem, we introduce the Discrete Cosine Transform (DCT) to reduce the correlations among the network inputs. As expected, experimental results reveal that DCT preprocessing substantially reduces the observed low-frequency preference in TSF. These results highlight input autocorrelation as a key factor shaping spectral bias in KANs for time series forecasting.

1 INTRODUCTION

Time series forecasting (TSF) supports applications ranging from financial markets and weather prediction to energy management and traffic estimation (Praveen et al., 2026; Venkatachalam et al., 2023; Mystakidis et al., 2024; Shao et al., 2024). Deep forecasting models include MLPs, recurrent networks, Transformers, temporal convolutions, and multiscale architectures (Gheyas & Smith, 2009; Medsker et al., 2001; Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017; Zhou et al., 2021; Bai et al., 2018; Wu et al., 2023; Jin et al., 2024). Kolmogorov–Arnold Networks (KANs) (Liu et al., 2025) offer a different parameterization: they replace fixed node activations and scalar edge weights with learnable univariate functions on edges. Their local spline bases provide a flexible representation of nonlinear and oscillatory functions.

Spectral bias describes the tendency of neural networks to learn low-frequency components faster than high-frequency components (Rahaman et al., 2019; Hong et al., 2022). Existing KAN analysis provides an encouraging contrast to conventional MLPs: under independent inputs, the effective Hessian condition number of a shallow spline KAN can remain bounded independently of spline resolution (Wang et al., 2025). Increasing the grid resolution therefore need not introduce the severe optimization anisotropy associated with conventional networks. This favorable picture motivates KANs for tasks with rich spectral content, while recent studies also examine their behavior under noise and irregular sampling (Shen et al., 2025; Zeng et al., 2025).

The independent-input setting, however, does not describe ordinary forecasting windows. Inputs consist of lagged observations of the same process and are inherently autocorrelated (Bartholomew, 1971; Cryer, 1986). For smooth or persistent signals, this dependence creates strongly unequal variances along different directions in lag space. Existing empirical studies suggest that temporal structure can affect frequency learning (Sun et al., 2025; Ackaah-Gyasi et al., 2023), but the connection between input autocorrelation and the optimization spectrum of KANs requires a more explicit analysis. The central question is whether the favorable spectral behavior established for independent coordinates survives in this correlated-input setting.

We address this question through a leading-order Hessian analysis and controlled experiments. Under a covariance-transfer approximation for the spline-transformed inputs, the lag autocorrelation matrix enters the Hessian as a Kronecker factor. Both bounds on the effective Hessian condition

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.

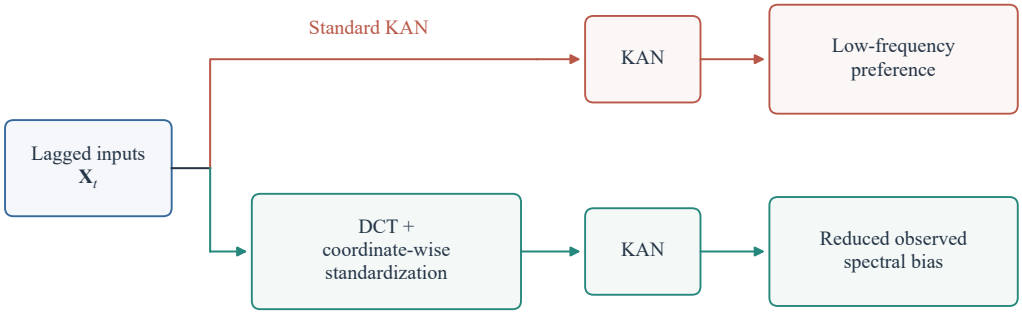


Figure 1: Overview of DCT-KAN. Autocorrelated lagged inputs can produce a low-frequency preference in KANs. An orthonormal DCT followed by training-fitted coordinate-wise standardization improves the recovery of difficult components.

number depend on its extreme eigenvalues. The analysis connects temporal dependence to unequal optimization rates, and predicts increased difficulty in directions associated with oscillatory target components. Synthetic forecasting experiments exhibit this behavior: stronger positive autocorrelation markedly impairs high-frequency recovery in the difficult lag directions.

We then apply the Discrete Cosine Transform (DCT) (Ahmed et al., 1974) to the lagged inputs, followed by coordinate-wise standardization. This preprocessing reduces cross-coordinate correlations and changes the scales presented to the learnable basis functions. DCT-KAN substantially improves the recovery of difficult components across the tested autoregressive configurations. Additional crossed frequency–direction experiments, swapped targets, and exact spline-Hessian calculations examine the mechanism beyond the original target construction. Figure 1 summarizes the resulting forecasting pipeline.

Our contributions are threefold:

- We identify temporal autocorrelation as a source of frequency- and direction-dependent learning difficulty in KAN forecasting, exposing a gap between independent-input theory and lagged-input practice.
- We derive effective condition-number bounds under a leading-order Hessian approximation, and quantify its matrix and spectral accuracy through direct B-spline calculations.
- We introduce DCT-KAN and demonstrate strong improvements on correlated synthetic tasks, supported by controlled frequency–direction comparisons and calibrated component recovery.

2 Background

2.1 KANs

KANs are inspired by the Kolmogorov–Arnold representation theorem (Kolmogorov, 1957; Arnold, 1957). Unlike an MLP, which learns scalar edge weights and applies fixed activations at nodes, a KAN learns univariate edge functions and sums their outputs. For architecture $[n_0, \ldots, n_L]$, layer l computes

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), \qquad j = 1, \ldots, n_{l+1}. \tag{1}$$

An edge function typically combines a residual activation, such as SiLU, with a learnable spline expansion. B-splines have local support: changing a coefficient affects a limited input interval, allowing finer grids to represent increasingly detailed functions. Our theoretical model isolates this spline component. For efficient forecasting experiments, we use FastKAN (Li, 2024), whose Gaussian RBF bases approximate cubic B-splines; the direct Hessian experiments use B-splines themselves.

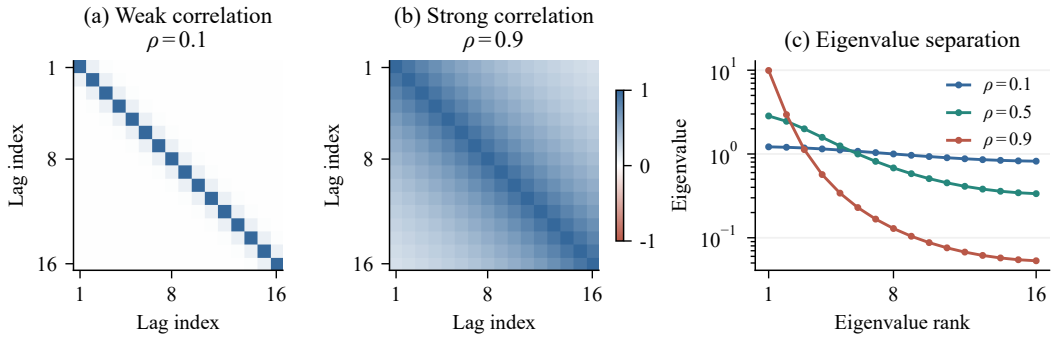


Figure 2: Input geometry for an AR(1) example with $p = 16$ and $R_{ij} = \rho^{|i-j|}$. Stronger autocorrelation increases off-diagonal correlations and separates the eigenvalues. The ranked eigenvalues describe variance along principal lag-space directions; their separation motivates the subsequent analysis of learning rates.

2.2 Spectral Bias and Optimization

Spectral bias is a difference in the learning rates of target components at different frequencies (Rahaman et al., 2019; Hong et al., 2022). If $\hat{f}_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ denotes a model's Fourier coefficient, the frequency-wise error is $e_{\boldsymbol{\omega}}(t) = |\hat{f}_{\boldsymbol{\theta}(t)}(\boldsymbol{\omega}) - \hat{f}(\boldsymbol{\omega})|$. A low-frequency preference occurs when this error decays faster for smaller frequency magnitudes.

The Hessian and neural tangent kernel (NTK) connect this phenomenon to optimization (Jacot et al., 2018; Arora et al., 2019). In a fixed linearization with squared loss, the parameter-space Gram matrix and the function-space NTK share their nonzero eigenvalues. For an error coefficient a_m along an eigenfunction with eigenvalue λ_m, gradient flow gives

$$a_m(t) = a_m(0) \exp(-\lambda_m t). \tag{2}$$

Small eigenvalues therefore correspond to slowly learned components. Fourier modes coincide with these eigenfunctions under suitable translation-invariant or periodic settings; more generally, the association between frequency and eigenvalue depends on the input distribution and kernel. A large condition number quantifies unequal learning rates, while the geometry of the corresponding eigenvectors determines which components are slow.

For a shallow KAN with fixed spline bases, the prediction is linear in its spline coefficients and the least-squares loss is quadratic. Under the independent-input setting of Wang et al. (2025), removing the constant-reparameterization directions yields

$$\kappa_{\mathrm{KAN}}(M) = \frac{\lambda_{\max}(M)}{\lambda^{+}_{\min}(M)} \leq C_k d, \tag{3}$$

where d is the input dimension and C_k depends on the spline degree rather than the grid resolution. This contrasts with the strong width dependence found for shallow ReLU MLPs (Wang et al., 2025; Hong et al., 2022). Local spline refinement can consequently improve representation without necessarily creating increasingly disparate optimization rates. The key issue for TSF is that input coordinates are correlated lags rather than independent samples.

Figure 2 illustrates how this dependence changes the input eigenspectrum. As autocorrelation grows, some combinations of lagged observations carry much more variance than others. The following analysis makes this geometry explicit in the spline Hessian and isolates the assumption needed to transfer it through nonlinear basis functions.

3 Spectral Bias of KANs in Time Series Forecasting

The preceding discussion shows that KANs are expected to be much less affected by spectral bias than MLPs when the input variables are statistically independent. However, this independence assumption is generally violated in time series forecasting. In a standard autoregressive forecasting setting, the input vector is constructed from consecutive lagged observations of the same time series, and hence

different input coordinates are naturally autocorrelated. In this section, we show that such temporal autocorrelation changes the Hessian spectrum of KAN training and reintroduces spectral bias.

3.1 Problem Setup and Notation

Consider a univariate time series $\{x_t\}_{t\in\mathbb{Z}}$. Given a look-back window of length p, the input vector is defined as

$$\mathbf{X}_t = [X_{t,1}, X_{t,2}, \ldots, X_{t,p}]^\top = [x_t, x_{t-1}, \ldots, x_{t-p+1}]^\top. \tag{4}$$

We assume a strictly stationary process with finite variance and standardized input coordinates. Let

$$R_{jj'} = \mathrm{Corr}(X_{t,j}, X_{t,j'}) = r(|j - j'|) \tag{5}$$

denote the autocorrelation matrix of the lagged input vector. For a stationary time series, R is a symmetric Toeplitz correlation matrix with $\mathrm{Tr}(R) = p$.

Following the standard spectral-bias analysis of KANs, we focus on a single-layer KAN and omit the residual SiLU term. Let

$$m = G + k \tag{6}$$

be the number of B-spline basis functions, where G is the number of knot intervals and k is the spline degree, using open clamped knots. The model is written as

$$\hat{y}_{t+1} = f_{\boldsymbol{\theta}}(\mathbf{X}_t) = \sum_{j=1}^{p}\sum_{\ell=1}^{m} c_{j\ell} B_\ell(X_{t,j}), \tag{7}$$

where $B_\ell(\cdot)$ denotes the ℓ-th B-spline basis function. The population MSE loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}\left[(y_{t+1} - f_{\boldsymbol{\theta}}(\mathbf{X}_t))^2\right]. \tag{8}$$

3.2 Leading-Order Hessian Approximation Under Temporal Autocorrelation

We describe temporal dependence through a covariance-transfer closure for the spline-transformed inputs. It separates two aspects of the Hessian: the marginal basis geometry, shared across stationary lag coordinates, and the coupling between coordinates, represented by the autocorrelation matrix. Their Kronecker structure makes the effect of input correlation on curvature analytically explicit. The difference between the exact transformed covariance and this closure defines a residual matrix, derived in Appendix A and measured directly in Appendix E. This decomposition links the analytical bounds to independently computed B-spline Hessians.

Define the B-spline basis vector

$$\mathbf{b}(z) = [B_1(z), B_2(z), \ldots, B_m(z)]^\top. \tag{9}$$

Let

$$\boldsymbol{\nu} = \mathbb{E}[\mathbf{b}(z)], \qquad C = \mathbb{E}[\mathbf{b}(z)\mathbf{b}(z)^\top], \qquad D = \boldsymbol{\nu}\boldsymbol{\nu}^\top. \tag{10}$$

Here, C is the Gram matrix of the B-spline basis functions, while $C - D$ is the covariance matrix of the centered basis vector.

Assumption 1 (Covariance-transfer closure)**.** *For analytical tractability, we approximate the cross-covariance matrix of the transformed lagged inputs by transferring the scalar autocorrelation coefficient of the original lagged variables to the covariance of the transformed basis vector:*

$$\mathrm{Cov}\left(\mathbf{b}(X_{t,j}), \mathbf{b}(X_{t,j'})\right) \approx R_{jj'}(C - D). \tag{11}$$

This approximation is exact for the diagonal blocks with $j = j'$, and is also consistent with the independent-input case where $R_{jj'} = 0$ for $j \neq j'$. For correlated inputs, it retains the marginal basis covariance and uses $R_{jj'}$ to describe cross-lag coupling; higher-order dependence is collected in the residual.

Proposition 1 (Leading-order Hessian under autocorrelated lagged inputs)**.** *Under Assumption 1, the population Hessian $M = \nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$ of the single-layer KAN admits the following leading-order block approximation:*

$$M_{jj'} \approx D + R_{jj'}(C - D), \qquad j, j' = 1, \ldots, p, \tag{12}$$

where each block $M_{jj'} \in \mathbb{R}^{m\times m}$. Equivalently,

$$M \approx J_p \otimes D + R \otimes (C - D), \tag{13}$$

where $J_p = \mathbf{1}_p \mathbf{1}_p^\top$ and $\otimes$ denotes the Kronecker product.

Proof. The detailed derivation is provided in Appendix A. □

Appendix E directly evaluates this approximation using fixed cubic B-spline Hessians across 240 settings. At $G = 4$ and $n = 10^5$, the mean relative Frobenius error is below 1.9% throughout the tested correlation and input-dimension grid; the same study quantifies the residual and spectral-tail differences at finer resolutions.

Proposition 1 provides a leading-order generalization of the Hessian structure used in the original KAN spectral-bias analysis. If the input coordinates are independent, then $R = I_p$, and (13) becomes

$$M \approx J_p \otimes D + I_p \otimes (C - D). \tag{14}$$

In this case, the diagonal blocks of M are C, while the off-diagonal blocks are D, which recovers the block structure used in the standard analysis of KANs (Wang et al., 2025). Therefore, the independent-input result is recovered as a special case of the leading-order approximation.

In time series forecasting, however, $R \neq I_p$ in general. The additional factor $R_{jj'}$ in (12) means that the Hessian spectrum is no longer determined only by the B-spline Gram matrix C, but is also shaped by the temporal autocorrelation matrix R.

3.3 Effective Condition Number

As in the standard KAN analysis, the parameterization contains degenerate directions due to the non-uniqueness of decomposing constant functions across different input coordinates. For the single-output case with p input lags, there are $p - 1$ such degenerate directions. We therefore define the effective condition number as

$$\kappa_{\mathrm{TSF}}(M) = \frac{\lambda_{\max}(M)}{\lambda_p(M)}, \tag{15}$$

where the eigenvalues are arranged in nondecreasing order and $\lambda_p(M)$ denotes the first non-degenerate eigenvalue.

Theorem 1 (Bounds under the leading-order Hessian approximation)**.** *Let R and the B-spline Gram matrix C be positive definite, with $m \geq 3$. In the following bounds, M denotes the leading-order matrix $J_p \otimes D + R \otimes (C - D)$ in Proposition 1:*

$$\frac{1}{2}\lambda_{\max}(R)\lambda_{\max}(C) \leq \lambda_{\max}(M) \leq p\lambda_{\max}(C), \tag{16}$$

and

$$\lambda_{\min}(R)\lambda_{\min}(C) \leq \lambda_p(M) \leq \lambda_{\min}(R)\lambda_{\max}(C). \tag{17}$$

Consequently,

$$\frac{\lambda_{\max}(R)}{2\lambda_{\min}(R)} \leq \kappa_{\mathrm{TSF}}(M) \leq \frac{p}{\lambda_{\min}(R)} C_k, \tag{18}$$

where

$$C_k = \frac{\lambda_{\max}(C)}{\lambda_{\min}(C)} \tag{19}$$

is the condition number associated with the B-spline Gram matrix (Wang et al., 2025).

Proof. The proof is given in Appendices B and C, where Appendix B derives the bounds for $\lambda_{\max}(M)$ and Appendix C derives the bounds for $\lambda_p(M)$. □

Theorem 1 shows that, within the leading-order Hessian approximation, the effective condition number of KAN training in TSF depends explicitly on the spectrum of the input autocorrelation matrix R. Compared with the independent-input case, temporal autocorrelation affects the Hessian spectrum through two mechanisms. First, the largest eigenvalue of M is lower-bounded by $\frac{1}{2}\lambda_{\max}(R)\lambda_{\max}(C)$, meaning that the dominant curvature direction becomes stronger when R has a large principal

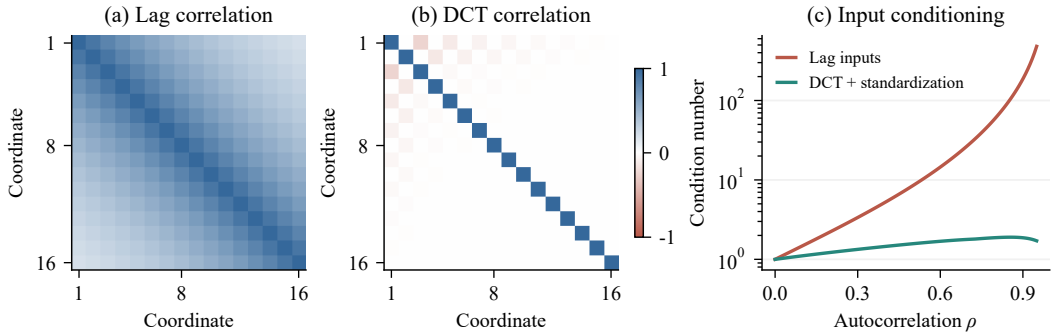


Figure 3: Analytical DCT illustration for the same $p = 16$ AR(1) family. (a,b) Correlation matrices before and after the orthonormal DCT at $\rho = 0.9$. (c) Condition numbers of the lag correlation matrix R and the standardized DCT covariance $SQRQ^\top S$, where $S_{ii} = (QRQ^\top)_{ii}^{-1/2}$. The DCT alone preserves covariance eigenvalues; the conditioning change shown here includes coordinate-wise standardization. All panels use analytical population matrices for the stated AR(1) process.

eigenvalue. Second, the first non-degenerate eigenvalue $\lambda_p(M)$ is controlled by $\lambda_{\min}(R)$, so a small minimum eigenvalue of R can substantially reduce the effective curvature in poorly conditioned directions.

When the lagged variables are independent, $R = I_p$, and hence

$$\lambda_{\max}(R) = \lambda_{\min}(R) = 1. \tag{20}$$

Then (18) reduces to

$$1 \le \kappa_{\mathrm{TSF}}(M) \le pC_k. \tag{21}$$

This is consistent with the standard KAN conclusion that the effective condition number grows at most linearly with the input dimension and is independent of the spline grid size G.

3.4 DCT Preprocessing

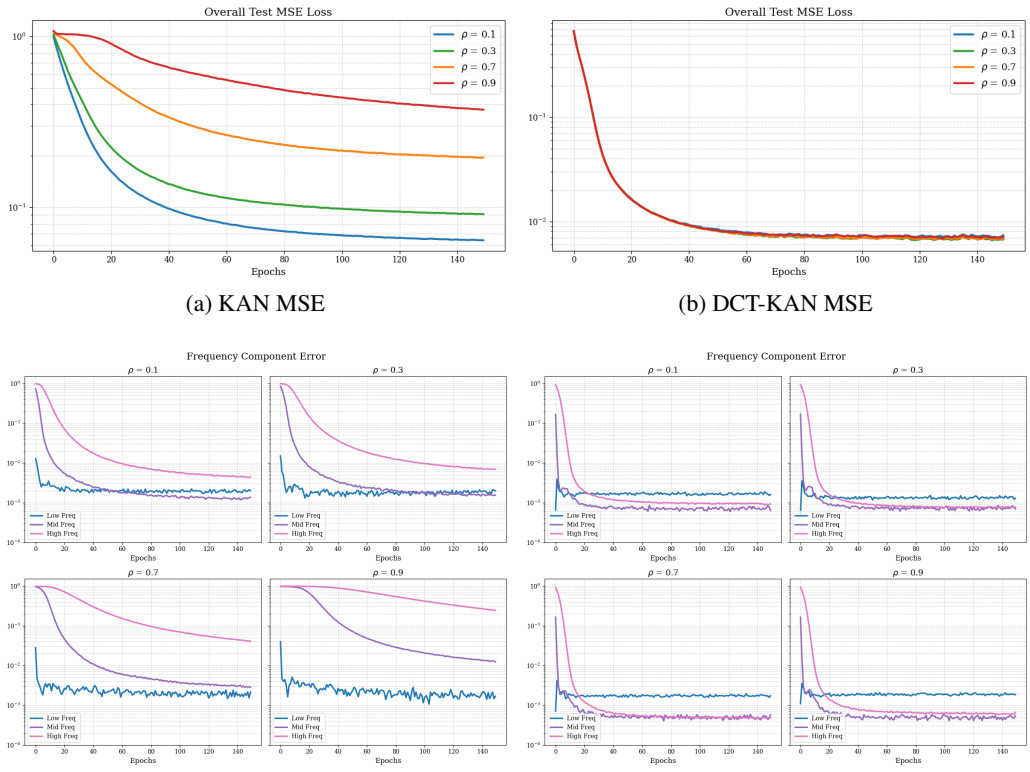
The condition-number bounds identify an optimization mechanism: large-eigenvalue directions are learned quickly, whereas small-eigenvalue directions are learned slowly. For persistent positively correlated inputs, the dominant lag-space directions are typically smooth, while weak-variance directions are more oscillatory. Autocorrelation can therefore amplify the difficulty of learning high-frequency components associated with those directions. The crossed experiments in Appendix D quantify this directional effect at fixed target frequencies, connecting the lag-space geometry to measured component recovery.

DCT-KAN changes the representation before the learnable edge functions are evaluated. Let Q be the orthonormal type-II DCT matrix, with indices $k, j = 0, \ldots, p-1$:

$$Q_{kj} = \alpha_k \cos\left[\frac{\pi}{p}\left(j + \frac{1}{2}\right)k\right], \quad \alpha_0 = p^{-1/2}, \quad \alpha_{k>0} = \sqrt{2/p}. \tag{22}$$

We transform the standardized lag vector as $\mathbf{z} = Q\widetilde{\mathbf{X}}$, then center and standardize each DCT coordinate using training-set statistics. All coefficients are retained, so the transformation does not discard input information. The DCT approximately separates correlated lag patterns, while subsequent standardization adjusts their scales before spline or RBF evaluation.

This distinction is visible in Figure 3. If Σ is the input covariance, the DCT alone gives $Q\Sigma Q^\top$, which has the same eigenvalues as Σ. With coordinate scaling $S = \mathrm{diag}(Q\Sigma Q^\top)^{-1/2}$, the standardized covariance becomes $SQ\Sigma Q^\top S$ and can have a much smaller condition number. Thus the practical method combines a change of coordinates with scale balancing. Its nonlinear learning behavior is evaluated directly in the following experiments, while Appendix D.5 examines the roles of target scaling and coordinate alignment.



(a) KAN MSE

(b) DCT-KAN MSE

(c) KAN Freq. Error

(d) DCT-KAN Freq. Error

Figure 4: Epoch-wise training dynamics of KAN and DCT-KAN ($N = 1$).

4 Experiments

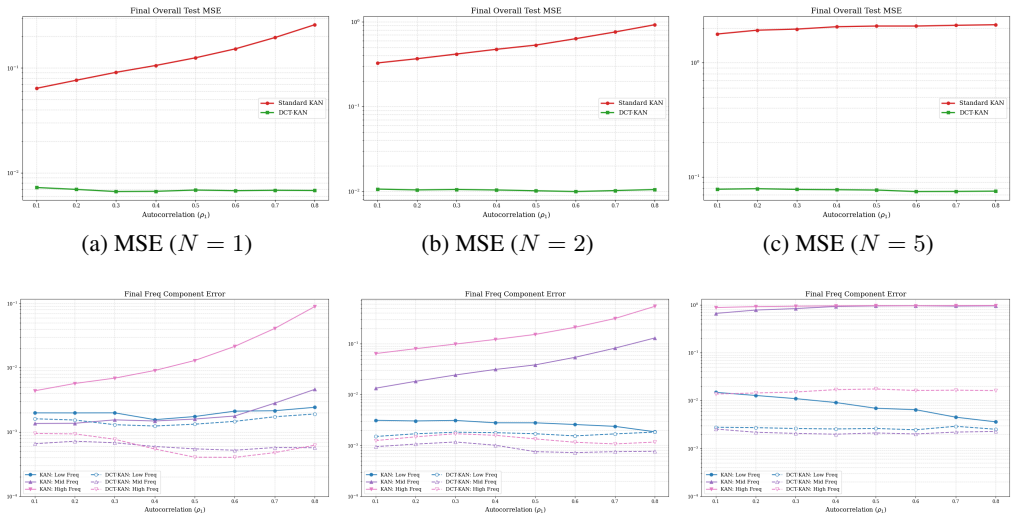
We test three predictions on controlled synthetic forecasting tasks: frequency-dependent learning under correlated inputs, stronger difficulty as autocorrelation increases, and improved component recovery with DCT preprocessing. Epoch-wise and autocorrelation-wise comparisons show how these effects combine across AR configurations. The crossed tests in Appendix D further separate frequency and direction, while Appendix E examines the Hessian approximation directly.

4.1 Experimental Overview

We compare KAN and DCT-KAN on synthetic AR(N) inputs for $(N, p) \in \{(1, 3), (2, 6), (5, 15)\}$. The target combines frequencies 1, 3, and 6 along standardized averaging and second-difference projections, allowing frequency-specific recovery to be tracked. Both methods use identical targets; DCT-KAN additionally applies an orthonormal DCT and training-fitted coordinate-wise standardization to its inputs. We report test MSE and component-wise recovery throughout training and across autocorrelation strengths. The full data-generating process, target construction, hyperparameters, and original recovery metric are provided in Appendix F. Appendix D further evaluates clean NMSE and calibrated joint component recovery under independently varied frequencies and directions.

4.2 KANs Exhibit Spectral Bias in Time Series Forecasting

We first examine the training dynamics of the standard KAN under different autocorrelation strengths. Figure 4a reports the epoch-wise test MSE for $N = 1$. It can be observed that, although the test MSE decreases for all values of ρ_1, the final converged MSE becomes increasingly large as ρ_1 increases. This indicates that stronger temporal autocorrelation makes the forecasting problem substantially more difficult for KANs. More importantly, the component-wise errors in Figure 4c reveal a clear



(a) MSE ($N = 1$) (b) MSE ($N = 2$) (c) MSE ($N = 5$)

(d) Freq. Error ($N = 1$) (e) Freq. Error ($N = 2$) (f) Freq. Error ($N = 5$)

Figure 5: Autocorrelation-wise performance of KAN and DCT-KAN.

frequency-dependent learning behavior. The low-frequency component is learned rapidly and its recovery error decreases quickly during training, whereas the middle- and high-frequency components converge much more slowly. When ρ_1 is large, the high-frequency error remains high even after sufficient training epochs, showing that KANs have difficulty recovering oscillatory components from strongly autocorrelated lagged inputs.

We further investigate this phenomenon by varying the autocorrelation strength more systematically. As shown in Figures 5a–5c, the test MSE of KAN consistently increases as ρ_1 grows. This trend is observed across all three AR configurations, suggesting that the degradation is not specific to a particular input dimension or AR order. The frequency-wise results in Figures 5d–5f provide a more detailed explanation: the recovery error of the low-frequency component remains relatively small, while the errors of the middle- and high-frequency components increase rapidly with ρ_1. Therefore, the performance degradation under stronger autocorrelation is mainly caused by the failure to learn higher-frequency components rather than by a uniform deterioration across all components.

The effect becomes even more pronounced as N increases. A larger N corresponds to a longer and more complex lagged input vector, which introduces richer correlation structures among input coordinates. Comparing the results for $N = 1$, $N = 2$, and $N = 5$ in Figure 5, we observe that the spectral bias of KANs becomes progressively stronger as the input sequence becomes more complex. In particular, when $N = 5$, KAN almost fails to recover the middle- and high-frequency components, as indicated by their persistently large component errors in Figure 5f. These empirical observations are consistent with the theoretical analysis in Section 3: stronger temporal autocorrelation makes the autocorrelation matrix R more ill-conditioned, enlarges the effective Hessian condition number, and consequently slows down optimization along directions associated with oscillatory high-frequency patterns.

4.3 DCT-KAN Alleviates Spectral Bias

We next examine whether decorrelating the lagged inputs can mitigate the spectral bias observed above. Figure 4b shows the epoch-wise MSE curves of DCT-KAN for $N = 1$ under different values of ρ_1. In contrast to the standard KAN in Figure 4a, the MSE curves of DCT-KAN are almost indistinguishable across different autocorrelation strengths. Moreover, the final converged MSE is substantially lower than that of the standard KAN, reaching approximately an order-of-magnitude improvement. This indicates that, after DCT preprocessing, the optimization and generalization behavior of KAN becomes much less sensitive to the temporal autocorrelation of the original lagged inputs.

The component-wise results further support this conclusion. As shown in Figure 4d, DCT-KAN learns the low-, middle-, and high-frequency components much more effectively than the standard KAN. In particular, the recovery errors of the middle- and high-frequency components decrease significantly during training and remain at a low level even when ρ_1 is large. Compared with Figure 4c, the gap between the low-frequency component and the higher-frequency components is greatly reduced. This suggests that DCT preprocessing weakens the low-frequency preference of KANs by reducing the correlation among input coordinates.

The autocorrelation-wise results in Figure 5 provide a more systematic comparison. Across all three AR configurations, DCT-KAN consistently achieves much smaller test MSE than the standard KAN, as shown in Figures 5a–5c. More importantly, its MSE remains nearly stable as ρ_1 increases, whereas the MSE of the standard KAN grows rapidly with stronger autocorrelation. The same pattern can be observed in the component-wise errors in Figures 5d–5f: DCT-KAN maintains low recovery errors for all frequency components, and the errors exhibit only weak dependence on ρ_1. Even in the most challenging case of $N = 5$, where the standard KAN almost fails to learn the middle- and high-frequency components, DCT-KAN still recovers these components effectively. These results confirm that the spectral bias observed in standard KANs for TSF is largely induced by the autocorrelation among lagged inputs, and that DCT preprocessing provides an effective way to alleviate this problem.

Additional crossed frequency–direction experiments in Appendix D strengthen the empirical comparison with 2,880 paired training runs. In the standardized-projection setting at $\rho = 0.9$, DCT-KAN reduces hard-direction test NMSE by approximately $62.7\times$ and $122.3\times$ at frequencies 3 and 6, respectively, and retains its advantage after swapping target assignments and adding fixed random phases. Joint component calibration and a fixed-physical-frequency comparison further distinguish the roles of target frequency, direction, and coordinate scaling.

These improvements also translate into faster recovery. For the two hard-direction targets at $\rho = 0.9$, all 30 DCT-KAN runs reach clean diagnostic NMSE at most 0.1, with median hitting times of 48 and 160 optimizer steps; none of the corresponding raw-input runs reaches this threshold within the training budget. At frequency 6, the raw-input NMSE is 0.0380 along the easy direction but 1.7292 along the hard direction. Holding frequency fixed thus exposes a substantial directional learning gap, directly connecting the forecasting results to the anisotropic optimization mechanism.

The complementary matrix experiments in Appendix E test the analytical approximation against explicitly constructed B-spline Hessians. At grid size $G = 4$ and sample size 10^5, the mean relative Frobenius error remains below 1.9% across all tested dimensions and autocorrelation strengths. Together, the learning and matrix experiments support both the practical benefit of the transformed representation and the leading-order description of correlation-dependent curvature. Fine-grid corrections and broader target geometries provide concrete directions for extending this account.

5 Conclusion

In this paper, we pointed out that the favorable spectral behavior of KANs under independent inputs does not directly extend to time series forecasting, where lagged input variables are inherently autocorrelated. Through a leading-order Hessian analysis based on a covariance-transfer approximation, we showed that temporal autocorrelation can alter the Hessian spectrum of KAN training and make the effective condition number depend on the extreme eigenvalues of the input autocorrelation matrix. As the autocorrelation strength increases, the optimization landscape becomes more anisotropic, causing KANs to learn low-frequency components much faster than middle- and high-frequency components. Controlled experiments on synthetic time series supported this prediction and showed that the spectral bias becomes increasingly severe under stronger autocorrelation. To address this issue, we introduced DCT preprocessing to reduce correlations among lagged inputs. The resulting DCT-KAN substantially alleviates the low-frequency preference and improves the recovery of high-frequency components, indicating that the spectral bias observed in KANs for time series forecasting is mainly induced by input autocorrelation rather than being an intrinsic limitation of the KAN architecture. More broadly, these findings identify the lag-space representation as a practical design choice for KAN forecasting: balancing correlated coordinates can make oscillatory target components substantially more accessible to optimization without discarding input information.

AI USE STATEMENT

Generative AI tools, including ChatGPT, were used to assist with the refinement of theoretical formulations and mathematical presentation, experimental design, software implementation, and interpretation of experimental results. They were also used to assist with literature search and summarization, drafting and editing parts of the manuscript, improving readability, and preparing research code and scientific figures. All AI-assisted mathematical derivations, claims, references, code, and written content were independently reviewed, verified, and revised by the authors. The generated and modified code was tested by the authors, and all reported experimental results were obtained from experiments conducted and evaluated by the authors rather than generated by AI. The authors take full responsibility for the methodology, results, claims, and final content of this work.

REPRODUCIBILITY STATEMENT

Section 3 states the theoretical assumptions, with proofs in Appendices A–C. Appendix F specifies the synthetic data generation, preprocessing, model configurations, and evaluation metrics. Appendices D and E detail the additional experimental protocols, random-seed design, statistical summaries, and implementation checks.

REFERENCES

Kofi Nketia Ackaah-Gyasi, Sergio Valdez, Yifeng Gao, and Li Zhang. Exploring spectral bias in time series long sequence forecasting. 2023.

Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

Vladimir I. Arnold. On functions of three variables. *Doklady Akademii Nauk SSSR*, 114(4):679–681, 1957.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 3 2018. URL https://arxiv.org/abs/1803.01271.

David J Bartholomew. Time series analysis forecasting and control., 1971.

Jonathan D Cryer. *Time series analysis*, volume 286. Duxbury Press Boston, 1986.

Iffat A. Gheyas and Leslie S. Smith. A neural network approach to time series forecasting. 2009. URL https://api.semanticscholar.org/CorpusID:2266156.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term memory. *Neural Computation*, 9(8): 1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735.

Qingguo Hong, Jonathan W. Siegel, Qinyang Tan, and Jinchao Xu. On the Activation Function Dependence of the Spectral Bias of Neural Networks, 8 2022. URL https://arxiv.org/abs/2208.04924.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Andrey N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114(5):953–956, 1957.

Ziyao Li. Kolmogorov-Arnold Networks are Radial Basis Function Networks, 5 2024. URL https://arxiv.org/abs/2405.06721.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Larry R Medsker, Lakhmi Jain, et al. Recurrent neural networks. *Design and applications*, 5(64-67): 2, 2001.

Aristeidis Mystakidis, Paraskevas Koukaras, Nikolaos Tsalikidis, Dimosthenis Ioannidis, and Christos Tjortjis. Energy forecasting: A comprehensive review of techniques and technologies. *Energies*, 17(7):1662, 2024.

Mande Praveen, Satish Dekka, Dasari Manendra Sai, Das Prakash Chennamsetty, and Durga Prasad Chinta. Financial time series forecasting: A comprehensive review of signal processing and optimization-driven intelligent models. *Computational Economics*, 67(2):963–989, 2026.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

Qitan Shao, Xinglin Piao, Xiangyu Yao, Yuqiu Kong, Yongli Hu, Baocai Yin, and Yong Zhang. An adaptive composite time series forecasting model for short-term traffic flow. *Journal of Big Data*, 11(1):102, 2024.

Haoran Shen, Chen Zeng, Jiahui Wang, and Qiao Wang. Reduced effectiveness of kolmogorov-arnold networks on functions with noise. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10890669.

Jialong Sun, Xinpeng Ling, Jiaxuan Zou, Jiawen Kang, and Kejia Zhang. FREIE: Low-Frequency spectral bias in neural networks for Time-Series tasks, 10 2025. URL https://arxiv.org/abs/2510.25800.

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

K Venkatachalam, Pavel Trojovskỳ, Dragan Pamucar, Nebojsa Bacanin, and Vladimir Simic. Dwfh: An improved data-driven deep weather forecasting hybrid model using transductive long short term memory (t-lstm). *Expert systems with applications*, 213:119270, 2023.

Yixuan Wang, Jonathan W Siegel, Ziming Liu, and Thomas Y Hou. On the expressiveness and spectral bias of kans. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Chen Zeng, Jiahui Wang, Haoran Shen, and Qiao Wang. Kan versus mlp on irregular or noisy function. In *2025 15th IEEE International Conference on Pattern Recognition Systems (ICPRS)*, pp. 1–7. IEEE, 2025.

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, 5 2021. doi: 10.1609/aaai.v35i12.17325.

A Derivation of Proposition 1

In this appendix, we derive the leading-order Hessian approximation used in Proposition 1. The derivation consists of two parts. First, we obtain an exact expression for the Hessian in terms of the cross-covariance matrices of the B-spline-transformed lagged inputs. Second, we introduce the covariance-transfer closure in Assumption 1. The resulting decomposition identifies the correlation-dependent leading term and the residual evaluated in Appendix E.

Recall that the single-layer KAN considered in the main text is

$$f_{\boldsymbol{\theta}}(\mathbf{X}_t) = \sum_{j=1}^{p} \sum_{\ell=1}^{m} c_{j\ell} B_\ell(X_{t,j}), \qquad m = G + k, \tag{23}$$

where p is the look-back window length and m is the number of B-spline basis functions. The population MSE loss is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}\left[(y_{t+1} - f_{\boldsymbol{\theta}}(\mathbf{X}_t))^2\right]. \tag{24}$$

Since the B-spline basis functions are bounded on the compact input domain considered in this paper, differentiation can be interchanged with expectation.

Let

$$e_t(\boldsymbol{\theta}) = y_{t+1} - f_{\boldsymbol{\theta}}(\mathbf{X}_t) \tag{25}$$

be the prediction error. Since the model in (23) is linear with respect to the spline coefficients $c_{j\ell}$, we have

$$\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{X}_t)}{\partial c_{j\ell}} = B_\ell(X_{t,j}). \tag{26}$$

Therefore,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_{j\ell}} &= \frac{\partial}{\partial c_{j\ell}} \frac{1}{2}\mathbb{E}\left[e_t(\boldsymbol{\theta})^2\right] \\
&= \mathbb{E}\left[e_t(\boldsymbol{\theta}) \frac{\partial e_t(\boldsymbol{\theta})}{\partial c_{j\ell}}\right] \\
&= -\mathbb{E}\left[(y_{t+1} - f_{\boldsymbol{\theta}}(\mathbf{X}_t))\, B_\ell(X_{t,j})\right] \\
&= \mathbb{E}\left[(f_{\boldsymbol{\theta}}(\mathbf{X}_t) - y_{t+1})\, B_\ell(X_{t,j})\right].
\end{aligned} \tag{27}$$

Taking another derivative with respect to $c_{j'\ell'}$ gives

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}}{\partial c_{j\ell} \partial c_{j'\ell'}} &= \mathbb{E}\left[\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{X}_t)}{\partial c_{j'\ell'}} B_\ell(X_{t,j})\right] \\
&= \mathbb{E}\left[B_\ell(X_{t,j}) B_{\ell'}(X_{t,j'})\right].
\end{aligned} \tag{28}$$

Thus, the Hessian is independent of the target value y_{t+1} and is determined only by the second-order statistics of the B-spline-transformed inputs.

Define the B-spline basis vector

$$\mathbf{b}(z) = [B_1(z), B_2(z), \ldots, B_m(z)]^\top. \tag{29}$$

For the j-th lagged input variable, denote

$$\mathbf{b}_j = \mathbf{b}(X_{t,j}). \tag{30}$$

Then the (j, j') block of the Hessian can be written exactly as

$$M_{jj'} = \mathbb{E}\left[\mathbf{b}_j \mathbf{b}_{j'}^\top\right], \tag{31}$$

where the (ℓ, ℓ') entry of $M_{jj'}$ is $\mathbb{E}[B_\ell(X_{t,j}) B_{\ell'}(X_{t,j'})]$.

By stationarity and marginal normalization, all lagged input coordinates are assumed to have the same marginal distribution. Hence, the mean vector

$$\boldsymbol{\nu} = \mathbb{E}[\mathbf{b}_j] \tag{32}$$

does not depend on j. Decompose each transformed input vector into its mean and centered fluctuation:

$$\mathbf{b}_j = \boldsymbol{\nu} + \widetilde{\mathbf{b}}_j, \qquad \mathbb{E}[\widetilde{\mathbf{b}}_j] = \mathbf{0}. \tag{33}$$

Substituting (33) into (31), we obtain

$$\begin{aligned} M_{jj'} &= \mathbb{E}\left[(\boldsymbol{\nu} + \widetilde{\mathbf{b}}_j)(\boldsymbol{\nu} + \widetilde{\mathbf{b}}_{j'})^\top\right] \\ &= \boldsymbol{\nu}\boldsymbol{\nu}^\top + \mathbb{E}\left[\widetilde{\mathbf{b}}_j \widetilde{\mathbf{b}}_{j'}^\top\right]. \end{aligned} \tag{34}$$

Define

$$D = \boldsymbol{\nu}\boldsymbol{\nu}^\top, \tag{35}$$

and

$$C = \mathbb{E}\left[\mathbf{b}_j \mathbf{b}_j^\top\right]. \tag{36}$$

Then the covariance matrix of the centered B-spline basis vector is

$$\begin{aligned} S &= \mathrm{Cov}(\mathbf{b}_j) \\ &= \mathbb{E}\left[(\mathbf{b}_j - \boldsymbol{\nu})(\mathbf{b}_j - \boldsymbol{\nu})^\top\right] \\ &= C - D. \end{aligned} \tag{37}$$

For $j = j'$, we have

$$M_{jj} = D + S = C. \tag{38}$$

For $j \neq j'$, define the exact cross-covariance matrix

$$\Gamma_{jj'} = \mathrm{Cov}\left(\mathbf{b}(X_{t,j}), \mathbf{b}(X_{t,j'})\right) = \mathbb{E}\left[\widetilde{\mathbf{b}}_j \widetilde{\mathbf{b}}_{j'}^\top\right]. \tag{39}$$

Then the exact Hessian block is

$$M_{jj'} = D + \Gamma_{jj'}. \tag{40}$$

Equation (40) is an exact identity. In general, however, for nonlinear basis functions $B_\ell(\cdot)$, the matrix $\Gamma_{jj'}$ is not determined solely by the scalar correlation coefficient

$$R_{jj'} = \mathrm{Corr}(X_{t,j}, X_{t,j'}). \tag{41}$$

It also depends on the full joint distribution of $(X_{t,j}, X_{t,j'})$ and on the nonlinear shape of the basis functions.

The covariance-transfer approximation used in Proposition 1 replaces the exact cross-covariance matrix $\Gamma_{jj'}$ by a scalar-correlation rescaling of the marginal covariance matrix $S = C - D$:

$$\Gamma_{jj'} \approx R_{jj'}(C - D). \tag{42}$$

Equivalently, one may write the exact cross-covariance as

$$\Gamma_{jj'} = R_{jj'}(C - D) + E_{jj'}, \tag{43}$$

where

$$E_{jj'} = \Gamma_{jj'} - R_{jj'}(C - D) \tag{44}$$

is the residual matrix associated with the covariance-transfer approximation. By construction, $E_{jj} = 0$, because $R_{jj} = 1$ and $\Gamma_{jj} = C - D$.

Thus, the exact Hessian can be written as

$$M_{jj'} = D + R_{jj'}(C - D) + E_{jj'}. \tag{45}$$

Let E denote the block matrix whose (j, j') block is $E_{jj'}$. Then the exact Hessian admits the decomposition

$$M = J_p \otimes D + R \otimes (C - D) + E, \tag{46}$$

where $J_p = \mathbf{1}_p \mathbf{1}_p^\top$.

The approximation in Proposition 1 corresponds to neglecting the residual matrix E, namely,

$$E \approx 0. \tag{47}$$

Under this covariance-transfer approximation, (45) reduces to

$$M_{jj'} \approx D + R_{jj'}(C - D), \qquad j, j' = 1, \ldots, p, \tag{48}$$

and the full Hessian becomes

$$M \approx J_p \otimes D + R \otimes (C - D). \tag{49}$$

This is the Kronecker-product representation stated in Proposition 1.

Equation (42) closes the second-moment description: the shape of the cross-covariance matrix of the transformed variables is approximated by the marginal covariance matrix $C - D$, while its magnitude is controlled by the original lag correlation coefficient $R_{jj'}$. The approximation is exact on the diagonal blocks. It is also consistent with the independent-input case: if $X_{t,j}$ and $X_{t,j'}$ are independent for $j \neq j'$, then $\Gamma_{jj'} = 0$ and $R_{jj'} = 0$, so the off-diagonal residuals vanish.

If one keeps the residual term, then the approximation error is explicitly represented by E in (46). In particular, if $\|E\|_2 \leq \varepsilon$, then Weyl's inequality gives

$$|\lambda_i(M) - \lambda_i\left(J_p \otimes D + R \otimes (C - D)\right)| \leq \varepsilon \tag{50}$$

for every eigenvalue index i. The residual norm thus directly controls the absolute spectral error and connects the closure to the matrix diagnostics in Appendix E.

A.1 Consistency of the Condition-Number Interval

The bounds in (18) are consistent with each other. Since R is a correlation matrix, its diagonal entries are all equal to one, and thus

$$\mathrm{Tr}(R) = p. \tag{51}$$

Therefore, for positive semidefinite R, we have

$$\lambda_{\max}(R) \leq p. \tag{52}$$

Moreover, since $C_k \geq 1$, it follows that

$$\frac{\lambda_{\max}(R)}{2\lambda_{\min}(R)} \leq \frac{p}{\lambda_{\min}(R)} C_k. \tag{53}$$

Hence, the lower bound in (18) does not exceed the upper bound.

B Bounds for $\lambda_{\max}(M)$

In this appendix, we derive the upper and lower bounds of the largest eigenvalue of the Hessian matrix M used in Theorem 1. Recall from Proposition 1 that, under the covariance-transfer approximation, the Hessian has the Kronecker-product form

$$M = J_p \otimes D + R \otimes (C - D), \tag{54}$$

where J_p is the $p \times p$ all-one matrix, R is the autocorrelation matrix of the lagged inputs, $C = \mathbb{E}[\mathbf{b}(z)\mathbf{b}(z)^{\mathrm{T}}]$, and $D = \boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}$ with $\boldsymbol{\nu} = \mathbb{E}[\mathbf{b}(z)]$.

For notational simplicity, define

$$S = C - D. \tag{55}$$

Since S is the covariance matrix of the centered B-spline basis vector, it is positive semidefinite. Thus,

$$M = J_p \otimes D + R \otimes S. \tag{56}$$

B.1 Upper Bound of $\lambda_{\max}(M)$

Both R and J_p are positive semidefinite with trace p, so $R \preceq pI_p$ and $J_p \preceq pI_p$. Since $D \succeq 0$ and $S = C - D \succeq 0$, the Kronecker product preserves these inequalities:

$$M = J_p \otimes D + R \otimes S \preceq pI_p \otimes D + pI_p \otimes S = pI_p \otimes C. \tag{57}$$

Consequently, every vector $\mathbf{z} \in \mathbb{R}^{pm}$ satisfies

$$\mathbf{z}^{\top} M \mathbf{z} \leq p\lambda_{\max}(C)\|\mathbf{z}\|_2^2. \tag{58}$$

The Rayleigh–Ritz theorem gives

$$\lambda_{\max}(M) \leq p\lambda_{\max}(C). \tag{59}$$

This argument applies directly to the leading-order matrix used throughout the theorem.

B.2 Lower Bound of $\lambda_{\max}(M)$

We next derive a lower bound that does not rely on the entrywise nonnegativity of R. Since both D and S are positive semidefinite, and since both J_p and R are also positive semidefinite, the two Kronecker-product terms in (56) are positive semidefinite. Hence, removing either term cannot increase the largest eigenvalue. Therefore,

$$\lambda_{\max}(M) \geq \lambda_{\max}(J_p \otimes D). \tag{60}$$

Because the largest eigenvalue of J_p is p, we have

$$\lambda_{\max}(J_p \otimes D) = p\lambda_{\max}(D). \tag{61}$$

Thus,

$$\lambda_{\max}(M) \geq p\lambda_{\max}(D). \tag{62}$$

Similarly,

$$\lambda_{\max}(M) \geq \lambda_{\max}(R \otimes S). \tag{63}$$

Since the eigenvalues of a Kronecker product are pairwise products of the eigenvalues of its two factors, we obtain

$$\lambda_{\max}(R \otimes S) = \lambda_{\max}(R)\lambda_{\max}(S). \tag{64}$$

Therefore,

$$\lambda_{\max}(M) \geq \lambda_{\max}(R)\lambda_{\max}(S). \tag{65}$$

Now, using (52), the bound in (62) implies

$$\lambda_{\max}(M) \geq \lambda_{\max}(R)\lambda_{\max}(D). \tag{66}$$

Combining (66) and (65), we get

$$2\lambda_{\max}(M) \geq \lambda_{\max}(R)\left(\lambda_{\max}(D) + \lambda_{\max}(S)\right). \tag{67}$$

Since

$$C = D + S, \tag{68}$$

the largest eigenvalue of C satisfies

$$\lambda_{\max}(C) \leq \lambda_{\max}(D) + \lambda_{\max}(S). \tag{69}$$

Substituting (69) into (67) yields

$$2\lambda_{\max}(M) \geq \lambda_{\max}(R)\lambda_{\max}(C). \tag{70}$$

Hence,

$$\lambda_{\max}(M) \geq \frac{1}{2}\lambda_{\max}(R)\lambda_{\max}(C). \tag{71}$$

Combining (59) and (71), we obtain

$$\frac{1}{2}\lambda_{\max}(R)\lambda_{\max}(C) \leq \lambda_{\max}(M) \leq p\lambda_{\max}(C), \tag{72}$$

which completes the proof.

This result shows that, even without assuming entrywise positive autocorrelation, the largest curvature of the KAN training objective is still controlled by the largest eigenvalue of the input autocorrelation matrix R. Therefore, when the lagged-input correlation structure becomes more concentrated in a dominant direction, the largest eigenvalue of the Hessian also increases at least proportionally up to a universal constant factor.

C BOUNDS FOR $\lambda_p(M)$

In this appendix, we derive the lower and upper bounds of the first non-degenerate eigenvalue $\lambda_p(M)$ used in Theorem 1. Throughout this appendix, M denotes the leading-order Hessian under the covariance-transfer approximation in Proposition 1, namely

$$M = J_p \otimes D + R \otimes (C - D), \tag{73}$$

where $J_p = \mathbf{1}_p \mathbf{1}_p^\top$, R is the autocorrelation matrix of the lagged inputs, C is the B-spline Gram matrix, and $D = \boldsymbol{\nu}\boldsymbol{\nu}^\top$ with

$$\boldsymbol{\nu} = \mathbb{E}[\mathbf{b}(z)]. \tag{74}$$

For notational simplicity, define

$$S = C - D. \tag{75}$$

Since S is the covariance matrix of the centered B-spline basis vector, it is positive semidefinite.

We first clarify the degenerate directions. B-spline basis functions satisfy the partition-of-unity property. Therefore, there exists a unit vector $\mathbf{v}_0 \in \mathbb{R}^m$, corresponding to the constant spline direction, such that

$$\mathbf{v}_0^\top \mathbf{b}(z) = \text{constant} \tag{76}$$

for all z in the spline domain. Consequently,

$$S\mathbf{v}_0 = \mathbf{0}. \tag{77}$$

Let

$$\mathcal{V}_\perp = \left\{ \mathbf{v} \in \mathbb{R}^m : \mathbf{v}^\top \mathbf{v}_0 = 0 \right\} \tag{78}$$

be the Euclidean orthogonal complement of the constant spline direction. We assume that S is positive definite on $\mathcal{V}_\perp$, which is the standard non-degeneracy condition for the spline Gram matrix.

The degenerate null space of M is

$$\mathcal{Z} = \left\{ \mathbf{a} \otimes \mathbf{v}_0 : \mathbf{a}^\top \mathbf{1}_p = 0 \right\}. \tag{79}$$

Indeed, for any $\mathbf{a}^\top \mathbf{1}_p = 0$, we have

$$(J_p \otimes D)(\mathbf{a} \otimes \mathbf{v}_0) = (J_p \mathbf{a}) \otimes (D\mathbf{v}_0) = \mathbf{0}, \tag{80}$$

and

$$(R \otimes S)(\mathbf{a} \otimes \mathbf{v}_0) = (R\mathbf{a}) \otimes (S\mathbf{v}_0) = \mathbf{0}. \tag{81}$$

Thus, $\dim(\mathcal{Z}) = p - 1$. The first non-degenerate eigenvalue $\lambda_p(M)$ is the smallest eigenvalue of M restricted to

$$\mathcal{W} = \mathcal{Z}^\perp. \tag{82}$$

Every vector $\mathbf{w} \in \mathcal{W}$ can be written as

$$\mathbf{w} = \alpha\, \mathbf{1}_p \otimes \mathbf{v}_0 + \mathbf{z}, \tag{83}$$

where $\alpha \in \mathbb{R}$ and

$$\mathbf{z} = [\mathbf{z}_1^\top, \ldots, \mathbf{z}_p^\top]^\top, \qquad \mathbf{z}_j \in \mathcal{V}_\perp. \tag{84}$$

Since $\mathbf{v}_0$ is orthogonal to $\mathcal{V}_\perp$, we have

$$\|\mathbf{w}\|_2^2 = \alpha^2 p + \sum_{j=1}^{p} \|\mathbf{z}_j\|_2^2. \tag{85}$$

C.1 LOWER BOUND OF $\lambda_p(M)$

Let

$$r_{\min} = \lambda_{\min}(R), \qquad \lambda_C = \lambda_{\min}(C). \tag{86}$$

Since R is positive definite, $r_{\min} > 0$. Moreover, because R is a correlation matrix, $\mathrm{Tr}(R) = p$, and hence $r_{\min} \le 1$.

For any $\mathbf{w} \in \mathcal{W}$ with the decomposition in (83), the quadratic form of M satisfies

$$\begin{aligned}\mathbf{w}^\top M\mathbf{w} &= \mathbf{w}^\top (J_p \otimes D)\mathbf{w} + \mathbf{w}^\top (R \otimes S)\mathbf{w} \\ &\geq \mathbf{w}^\top (J_p \otimes D)\mathbf{w} + r_{\min}\, \mathbf{w}^\top (I_p \otimes S)\mathbf{w},\end{aligned} \tag{87}$$

where we used $R \succeq r_{\min} I_p$ and $S \succeq 0$.

Because $S\mathbf{v}_0 = \mathbf{0}$, the second term depends only on the non-constant components:

$$\mathbf{w}^\top (I_p \otimes S)\mathbf{w} = \sum_{j=1}^{p} \mathbf{z}_j^\top S\mathbf{z}_j. \tag{88}$$

For the first term, using $D = \boldsymbol{\nu}\boldsymbol{\nu}^\top$, we have

$$\begin{aligned}\mathbf{w}^\top (J_p \otimes D)\mathbf{w} &= \left(\sum_{j=1}^{p} (\alpha\mathbf{v}_0 + \mathbf{z}_j)\right)^\top D \left(\sum_{j=1}^{p} (\alpha\mathbf{v}_0 + \mathbf{z}_j)\right) \\ &= \left[\boldsymbol{\nu}^\top \left(p\alpha\mathbf{v}_0 + \sum_{j=1}^{p} \mathbf{z}_j\right)\right]^2.\end{aligned} \tag{89}$$

Define

$$\bar{\mathbf{z}} = \frac{1}{p}\sum_{j=1}^{p} \mathbf{z}_j, \qquad \boldsymbol{\delta}_j = \mathbf{z}_j - \bar{\mathbf{z}}. \tag{90}$$

Then

$$\sum_{j=1}^{p} \boldsymbol{\delta}_j = \mathbf{0}, \qquad \bar{\mathbf{z}} \in \mathcal{V}_\perp, \qquad \boldsymbol{\delta}_j \in \mathcal{V}_\perp. \tag{91}$$

Furthermore,

$$\sum_{j=1}^{p} \|\mathbf{z}_j\|_2^2 = p\|\bar{\mathbf{z}}\|_2^2 + \sum_{j=1}^{p} \|\boldsymbol{\delta}_j\|_2^2, \tag{92}$$

and

$$\sum_{j=1}^{p} \mathbf{z}_j^\top S\mathbf{z}_j = p\bar{\mathbf{z}}^\top S\bar{\mathbf{z}} + \sum_{j=1}^{p} \boldsymbol{\delta}_j^\top S\boldsymbol{\delta}_j. \tag{93}$$

Substituting (89) and (93) into (87), we obtain

$$\mathbf{w}^\top M\mathbf{w} \geq p^2 \left[\boldsymbol{\nu}^\top(\alpha\mathbf{v}_0 + \bar{\mathbf{z}})\right]^2 + r_{\min} p\bar{\mathbf{z}}^\top S\bar{\mathbf{z}} + r_{\min} \sum_{j=1}^{p} \boldsymbol{\delta}_j^\top S\boldsymbol{\delta}_j. \tag{94}$$

Since $r_{\min} \leq 1 \leq p$, we have

$$\begin{aligned}&p^2 \left[\boldsymbol{\nu}^\top(\alpha\mathbf{v}_0 + \bar{\mathbf{z}})\right]^2 + r_{\min} p\bar{\mathbf{z}}^\top S\bar{\mathbf{z}} \\ &\geq r_{\min} p \left\{\bar{\mathbf{z}}^\top S\bar{\mathbf{z}} + \left[\boldsymbol{\nu}^\top(\alpha\mathbf{v}_0 + \bar{\mathbf{z}})\right]^2\right\}.\end{aligned} \tag{95}$$

Because $S\mathbf{v}_0 = \mathbf{0}$ and $D = \boldsymbol{\nu}\boldsymbol{\nu}^\top$, the expression in braces is exactly the quadratic form of $C = S + D$ along the vector $\alpha\mathbf{v}_0 + \bar{\mathbf{z}}$:

$$\begin{aligned}\bar{\mathbf{z}}^\top S\bar{\mathbf{z}} + \left[\boldsymbol{\nu}^\top(\alpha\mathbf{v}_0 + \bar{\mathbf{z}})\right]^2 &= (\alpha\mathbf{v}_0 + \bar{\mathbf{z}})^\top C(\alpha\mathbf{v}_0 + \bar{\mathbf{z}}) \\ &\geq \lambda_C \left(\alpha^2 + \|\bar{\mathbf{z}}\|_2^2\right),\end{aligned} \tag{96}$$

where we used $\mathbf{v}_0^\top \bar{\mathbf{z}} = 0$.

It remains to control the deviation components. Since S is positive definite on $\mathcal{V}_\perp$, define

$$s_\perp = \min_{\mathbf{v} \in \mathcal{V}_\perp,\, \|\mathbf{v}\|_2 = 1} \mathbf{v}^\top S\mathbf{v}. \tag{97}$$

The matrix $C = S + D$ is a rank-one positive semidefinite perturbation of S. Since S has one zero eigenvalue associated with the constant direction $\mathbf{v}_0$, the Cauchy interlacing theorem implies

$$\lambda_C = \lambda_{\min}(C) \le s_\perp. \tag{98}$$

Therefore, for every $\boldsymbol{\delta}_j \in \mathcal{V}_\perp$,

$$\boldsymbol{\delta}_j^\top S \boldsymbol{\delta}_j \ge \lambda_C \|\boldsymbol{\delta}_j\|_2^2. \tag{99}$$

Combining (94), (95), (96), and (99), we obtain

$$\begin{aligned}
\mathbf{w}^\top M \mathbf{w} &\ge r_{\min} p \lambda_C \left(\alpha^2 + \|\bar{\mathbf{z}}\|_2^2\right) + r_{\min} \lambda_C \sum_{j=1}^{p} \|\boldsymbol{\delta}_j\|_2^2 \\
&= r_{\min} \lambda_C \left(\alpha^2 p + p\|\bar{\mathbf{z}}\|_2^2 + \sum_{j=1}^{p} \|\boldsymbol{\delta}_j\|_2^2 \right) \\
&= r_{\min} \lambda_C \|\mathbf{w}\|_2^2,
\end{aligned} \tag{100}$$

where the last equality follows from (85) and (92).

Thus, for every nonzero $\mathbf{w} \in \mathcal{W}$,

$$\frac{\mathbf{w}^\top M \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \ge \lambda_{\min}(R) \lambda_{\min}(C). \tag{101}$$

Since $\lambda_p(M)$ is the smallest eigenvalue of M on the non-degenerate subspace $\mathcal{W}$, the Rayleigh–Ritz theorem gives

$$\lambda_p(M) \ge \lambda_{\min}(R) \lambda_{\min}(C). \tag{102}$$

C.2 Upper Bound of $\lambda_p(M)$

We next derive an upper bound using a test direction in the non-degenerate subspace. Let $\mathbf{u}_{\min} \in \mathbb{R}^p$ be a unit eigenvector of R corresponding to $\lambda_{\min}(R)$:

$$R\mathbf{u}_{\min} = \lambda_{\min}(R)\mathbf{u}_{\min}, \qquad \|\mathbf{u}_{\min}\|_2 = 1. \tag{103}$$

We choose a unit vector $\mathbf{v} \in \mathcal{V}_\perp$ satisfying

$$\boldsymbol{\nu}^\top \mathbf{v} = 0. \tag{104}$$

Such a vector exists whenever the spline basis dimension is at least three, i.e., $m \ge 3$, or more generally whenever

$$\mathcal{V}_\perp \cap \boldsymbol{\nu}^\perp \tag{105}$$

is nontrivial. This is the usual setting for practical KANs, where the number of spline basis functions is larger than two.

Consider the test vector

$$\mathbf{w} = \mathbf{u}_{\min} \otimes \mathbf{v}. \tag{106}$$

Since $\mathbf{v} \in \mathcal{V}_\perp$, the vector $\mathbf{w}$ is orthogonal to the degenerate null space $\mathcal{Z}$ and hence belongs to $\mathcal{W} = \mathcal{Z}^\perp$. Moreover,

$$\|\mathbf{w}\|_2 = \|\mathbf{u}_{\min}\|_2 \|\mathbf{v}\|_2 = 1. \tag{107}$$

The Rayleigh quotient of M along this direction is

$$\begin{aligned}
\mathbf{w}^\top M \mathbf{w} &= (\mathbf{u}_{\min}^\top J_p \mathbf{u}_{\min})(\mathbf{v}^\top D \mathbf{v}) + (\mathbf{u}_{\min}^\top R \mathbf{u}_{\min})(\mathbf{v}^\top S \mathbf{v}) \\
&= (\mathbf{u}_{\min}^\top J_p \mathbf{u}_{\min})(\boldsymbol{\nu}^\top \mathbf{v})^2 + \lambda_{\min}(R) \mathbf{v}^\top S \mathbf{v}.
\end{aligned} \tag{108}$$

By the choice of $\mathbf{v}$ in (104), the first term vanishes. Hence

$$\mathbf{w}^\top M \mathbf{w} = \lambda_{\min}(R) \mathbf{v}^\top S \mathbf{v}. \tag{109}$$

Furthermore, since $\boldsymbol{\nu}^\top \mathbf{v} = 0$, we have

$$\mathbf{v}^\top D \mathbf{v} = (\boldsymbol{\nu}^\top \mathbf{v})^2 = 0, \tag{110}$$

and therefore

$$\mathbf{v}^\top S\mathbf{v} = \mathbf{v}^\top (C - D)\mathbf{v} = \mathbf{v}^\top C\mathbf{v}. \tag{111}$$

It follows that

$$\mathbf{w}^\top M\mathbf{w} = \lambda_{\min}(R)\mathbf{v}^\top C\mathbf{v} \leq \lambda_{\min}(R)\lambda_{\max}(C). \tag{112}$$

Since $\lambda_p(M)$ is the minimum Rayleigh quotient of M over the non-degenerate subspace $\mathcal{W}$, we have

$$\begin{aligned}\lambda_p(M) &= \min_{\mathbf{w}\in\mathcal{W},\, \|\mathbf{w}\|_2=1} \mathbf{w}^\top M\mathbf{w} \\ &\leq (\mathbf{u}_{\min} \otimes \mathbf{v})^\top M(\mathbf{u}_{\min} \otimes \mathbf{v}) \\ &\leq \lambda_{\min}(R)\lambda_{\max}(C).\end{aligned} \tag{113}$$

Therefore,

$$\lambda_p(M) \leq \lambda_{\min}(R)\lambda_{\max}(C). \tag{114}$$

Combining the lower bound in (102) and the upper bound in (113), we obtain

$$\lambda_{\min}(R)\lambda_{\min}(C) \leq \lambda_p(M) \leq \lambda_{\min}(R)\lambda_{\max}(C). \tag{115}$$

This completes the derivation of the bounds for the first non-degenerate eigenvalue of the Hessian matrix.

D DISENTANGLING FREQUENCY AND PROJECTION DIRECTION

The experiments in this appendix sharpen the empirical characterization of spectral bias by varying target frequency and input direction independently. Across 2,880 training runs, they identify the directions in which autocorrelation most strongly affects learning and demonstrate substantial improvements from DCT-KAN on the standardized-projection tasks. Swapped assignments, random phases, and calibrated component measurements provide complementary tests of this effect.

D.1 EXPERIMENTAL PROTOCOL

We generate stationary AR(1) sequences with $\rho \in \{0, 0.5, 0.9\}$, innovation standard deviation $\sqrt{1-\rho^2}$, and a burn-in of 1,000 samples. Each input contains $p = 6$ lags, ordered from newest to oldest. A continuous sequence supplies 4,000 training and 1,000 subsequent test windows; separate innovation streams supply 1,000 validation and 2,048 diagnostic windows. All preprocessing statistics are fitted on the training set. The two unit-norm directions are

$$\mathbf{w}_{\text{easy}} = \frac{1}{\sqrt{6}}[1,1,1,1,1,1]^\top, \qquad \mathbf{w}_{\text{hard}} = \frac{1}{\sqrt{12}}[1,-2,1,1,-2,1]^\top. \tag{116}$$

For the main crossed design, each direction is combined with every frequency $\omega \in \{1, 3, 6\}$. We standardize the projection $\mathbf{w}^\top \widetilde{\mathbf{X}}$ using its training mean and standard deviation, obtaining $V_{\mathbf{w}}$, and define a single-component target from $\sin(\omega V_{\mathbf{w}})$. Each clean component is centered and scaled to unit training variance. Independent Gaussian observation noise is added with variance equal to the clean target's training variance divided by 400. Both input representations use identical labels.

We compare Raw KAN with DCT-KAN, denoted *DCT+Std* in the figures. Both use FastKAN with architecture $[6, 50, 1]$, eight RBF grid points on $[-2, 2]$, the SiLU base-update branch enabled, and LayerNorm disabled. The DCT representation is an orthonormal type-II DCT of the standardized lag vector, followed by training-fitted coordinate-wise standardization. Adam uses learning rate 0.005, batch size 256, and 150 epochs, totaling 2,400 optimizer steps. Raw and DCT runs share initialization and minibatch order, while their trainable parameters are independent.

Each configuration uses ten independent data seeds and three paired initializations per seed. The main crossed design has 36 configurations; its fixed-physical-frequency counterpart has another 36; the mixed-target tests have 24, giving $96 \times 10 \times 3 = 2{,}880$ final runs. We report the final epoch. Statistical summaries first average the three initializations within each data seed and then compute a mean and two-sided 95% Student-t interval across the ten independent data seeds.

The primary metric is clean test normalized mean squared error,

$$\text{NMSE} = \frac{\frac{1}{n_{\text{test}}} \sum_i (\hat{y}_i - y_i^\star)^2}{\text{Var}_{\text{test}}(y^\star)}, \tag{117}$$

where $y^\star$ denotes the noise-free target. Learning speed is measured by the first diagnostic evaluation at which clean NMSE is at most 0.1, checked every epoch (16 optimizer steps). Runs that never reach this threshold are recorded as unreached.

D.2 CROSSED FREQUENCY–DIRECTION RESULTS

Table 1 reports the complete main crossed design. The strongest benefit appears precisely in the strongly correlated, oscillatory input direction emphasized by the analysis. At $\rho = 0.9$, the Raw KAN NMSE along the hard direction increases from 0.0272 to 0.4487 and 1.7292 as ω increases from 1 to 3 and 6. DCT-KAN reduces the latter two errors to 0.0072 and 0.0141, respectively: approximately $62.7\times$ and $122.3\times$ lower errors, corresponding to reductions of 98.4% and 99.2%. These comparisons hold the input direction fixed and therefore directly demonstrate the frequency dependence of the learning difficulty within that direction.

Table 1: Complete standardized-projection crossed design. Clean test NMSE is reported as mean ± 95% confidence-interval half-width over ten data seeds, after averaging three initializations per seed.

ρ	Direction	ω	Raw KAN	DCT-KAN
0	Easy	1	0.0081 ± 0.0019	0.0032 ± 0.0003
0	Easy	3	0.0868 ± 0.0056	0.0074 ± 0.0017
0	Easy	6	1.7403 ± 0.1177	0.0173 ± 0.0035
0	Hard	1	0.0053 ± 0.0010	0.0036 ± 0.0006
0	Hard	3	0.0336 ± 0.0041	0.0067 ± 0.0022
0	Hard	6	1.4292 ± 0.1388	0.0154 ± 0.0055
0.5	Easy	1	0.0056 ± 0.0007	0.0033 ± 0.0006
0.5	Easy	3	0.0489 ± 0.0076	0.0087 ± 0.0024
0.5	Easy	6	0.3171 ± 0.0256	0.0157 ± 0.0033
0.5	Hard	1	0.0106 ± 0.0020	0.0034 ± 0.0006
0.5	Hard	3	0.1667 ± 0.0211	0.0063 ± 0.0013
0.5	Hard	6	2.1628 ± 0.0434	0.0134 ± 0.0024
0.9	Easy	1	0.0022 ± 0.0004	0.0038 ± 0.0008
0.9	Easy	3	0.0075 ± 0.0023	0.0101 ± 0.0050
0.9	Easy	6	0.0380 ± 0.0100	0.0186 ± 0.0105
0.9	Hard	1	0.0272 ± 0.0067	0.0032 ± 0.0005
0.9	Hard	3	0.4487 ± 0.0408	0.0072 ± 0.0022
0.9	Hard	6	1.7292 ± 0.0534	0.0141 ± 0.0044

Figure 6 makes the corresponding optimization advantage explicit. At $\rho = 0.9$, none of the 30 Raw KAN runs reaches the NMSE threshold for hard-direction frequencies 3 and 6, whereas all 30 DCT-KAN runs do so, with median first-passage times of 48 and 160 steps. For DCT-KAN, these times match those at $\rho = 0$ and are the same for the two directions in the displayed conditions. Thus, on these standardized-projection tasks, preprocessing largely removes the strong directional disparity seen in Raw KAN.

At a fixed frequency, the crossed design also reveals the role of input geometry: for $\omega = 6$ and $\rho = 0.9$, Raw KAN has NMSE 0.0380 along the easy direction but 1.7292 along the hard direction. The result refines the mechanism to a joint dependence on frequency and direction. In the easy, low-frequency cases, Raw KAN is already accurate and preprocessing is not uniformly better; the substantial gains concentrate on the difficult components.

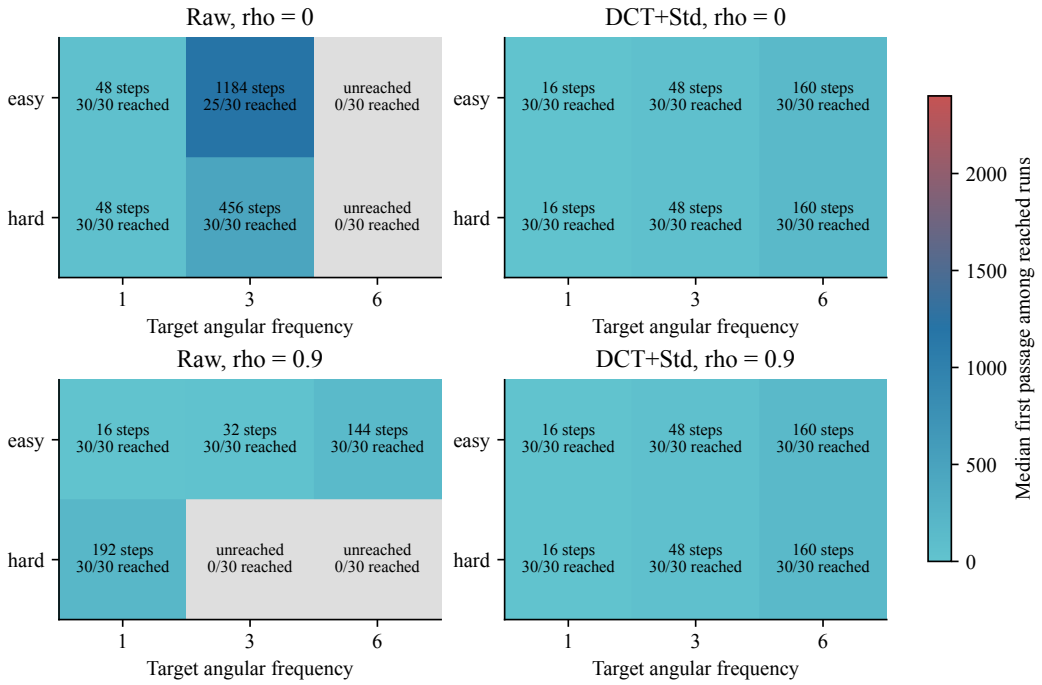


Figure 6: Frequency–direction learning-speed comparison for standardized projections at $\rho = 0$ and 0.9. Each cell reports the median optimizer steps *among runs reaching* diagnostic clean NMSE ≤ 0.1, together with the reached count out of 30. Gray cells indicate no successful run within 2,400 steps. Medians and reached counts jointly summarize recovery speed and reliability.

D.3 Swapped assignments and random phases

We next combine three unit-variance components. The original assignment places $\omega = 1$ on the easy direction and $\omega \in \{3, 6\}$ on the hard direction. The swapped assignment places $\omega \in \{1, 3\}$ on the hard direction and $\omega = 6$ on the easy direction. Each assignment is evaluated both with zero phases and with a single set of random phases drawn before training and held fixed across all data seeds and correlation levels.

Table 2 shows that DCT-KAN retains a clear advantage at $\rho = 0.9$ under all four combinations. With random phases, its clean NMSE is 0.0086 for the original assignment and 0.0098 for the swapped assignment, compared with 0.7165 and 0.1539 for Raw KAN. The corresponding improvements are approximately $83.6\times$ and $15.7\times$. Swapping the highest frequency to the easy direction improves Raw KAN, while DCT-KAN remains accurate across both assignments. The gain therefore extends beyond the original frequency assignment and the zero-phase target.

Table 2: Mixed-target clean test NMSE at $\rho = 0.9$. The uncertainty convention is the same as in Table 1. Random phases are fixed across data seeds.

Assignment / phase	Raw KAN	DCT-KAN
Original / zero	0.6866 ± 0.0414	0.0057 ± 0.0013
Original / random	0.7165 ± 0.0478	0.0086 ± 0.0030
Swapped / zero	0.1377 ± 0.0189	0.0109 ± 0.0033
Swapped / random	0.1539 ± 0.0216	0.0098 ± 0.0024

D.4 Calibrated component recovery

To resolve the contributions of correlated target components, we evaluate them jointly. Let F_c contain the centered clean components on the fixed diagnostic set, and let $\hat{\mathbf{y}}_c$ be the centered predictions. We

calculate

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\hat{\mathbf{y}}_c - F_c\mathbf{a}\|_2^2, \qquad E_q^{\text{joint}} = (\hat{a}_q - 1)^2, \tag{118}$$

and report the residual energy $\|\hat{\mathbf{y}}_c - F_c\hat{\mathbf{a}}\|_2^2/n$ together with total NMSE. The largest observed condition number of $F_c^\top F_c$ is 1.527, allowing ordinary least squares without ridge regularization. Coefficients remain unclipped.

Known-answer tests use zero prediction, perfect reconstruction, one missing component, and one doubled component. The joint metric recovers all prescribed coefficients to numerical precision: the largest component error for perfect reconstruction is below 5×10^{-29}. When one component is doubled, the error averaged over the three components is $1/3$, as expected, whereas the original clipped metric gives approximately 5.27×10^{-4}; the clipped metric also gives a maximum component error of 0.00951 for perfect reconstruction. These checks motivate using the joint metric as the primary component-level diagnostic in this appendix. Figure 7 shows that the DCT-KAN advantage remains evident with this calibrated measurement.

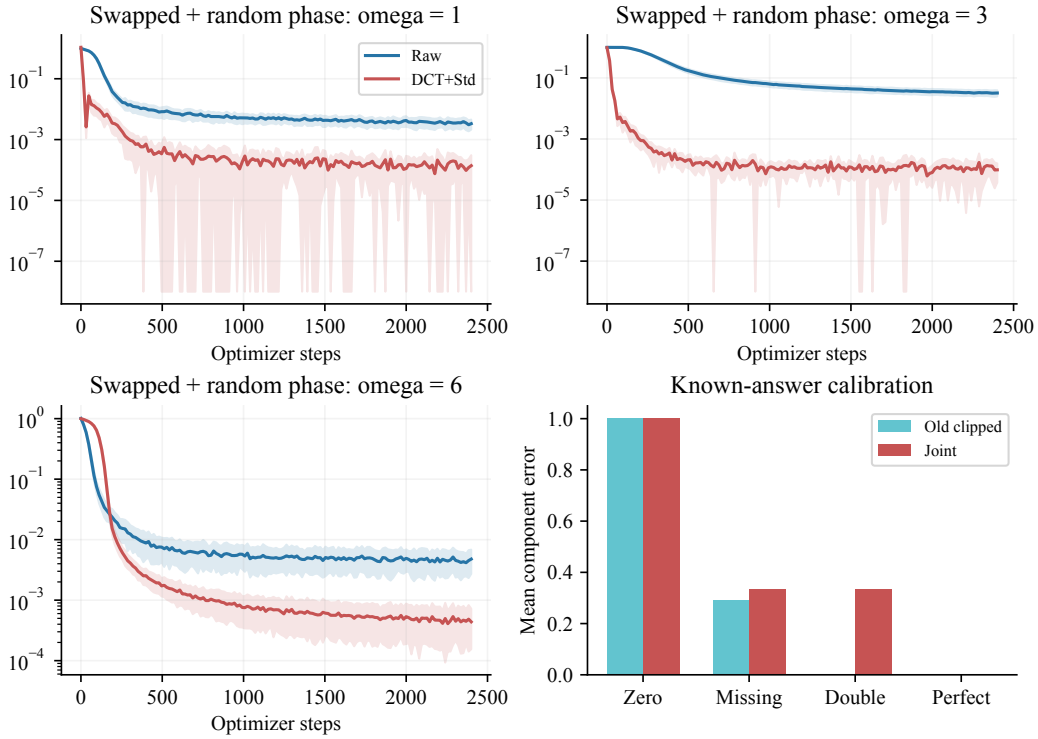


Figure 7: Swapped targets with fixed random phases and metric calibration. The first three panels show joint coefficient errors for $\rho = 0.9$, with $\omega = 1, 3$ on the hard direction and $\omega = 6$ on the easy direction. Lines are data-seed means after averaging initializations; shading denotes 95% Student-t intervals. Nonpositive lower interval limits are clipped to 10^{-8} only for logarithmic display. The last panel averages the known-answer component errors over all tested correlations, assignments, phases, and data seeds.

D.5 Fixed physical frequency and coordinate alignment

To separate projection normalization from a fixed target in the original lag coordinates, we additionally use $\sin(\omega \mathbf{w}^\top \mathbf{X})$ on the unstandardized physical lags. The frequency vector $\omega\mathbf{w}$ is fixed across ρ; only each component's output mean and variance are normalized. The model inputs retain the same training-fitted preprocessing. Figure 8 and Table 3 report this complementary experiment.

The two target conventions have distinct behavior. In particular, for fixed physical frequency along easy/$\omega = 6$ at $\rho = 0.9$, Raw KAN and DCT-KAN have NMSE 0.4443 and 1.9656, respectively, and neither reaches the diagnostic threshold. Moreover, the chosen easy and hard directions align with the DC and index-4 DCT basis directions for $p = 6$, and DCT-KAN also improves several tasks at

$\rho = 0$. Accordingly, the strong gains in the standardized-projection experiments reflect a combination of input geometry, coordinate alignment, and frequency scaling, rather than isolating decorrelation alone. Extending the comparison to arbitrary projection orientations and adaptive frequency scaling is a natural direction for future work.

Table 3: Fixed-physical-frequency sensitivity at $\rho = 0.9$: clean test NMSE with the same statistical convention as Table 1. All six direction–frequency combinations are included.

ρ	Direction	ω	Raw KAN	DCT-KAN
0.9	Easy	1	0.0044 ± 0.0014	0.0058 ± 0.0018
0.9	Easy	3	0.0482 ± 0.0108	0.0221 ± 0.0102
0.9	Easy	6	0.4443 ± 0.0782	1.9656 ± 0.2764
0.9	Hard	1	0.0113 ± 0.0032	0.0033 ± 0.0008
0.9	Hard	3	0.0196 ± 0.0043	0.0034 ± 0.0007
0.9	Hard	6	0.0698 ± 0.0166	0.0036 ± 0.0008

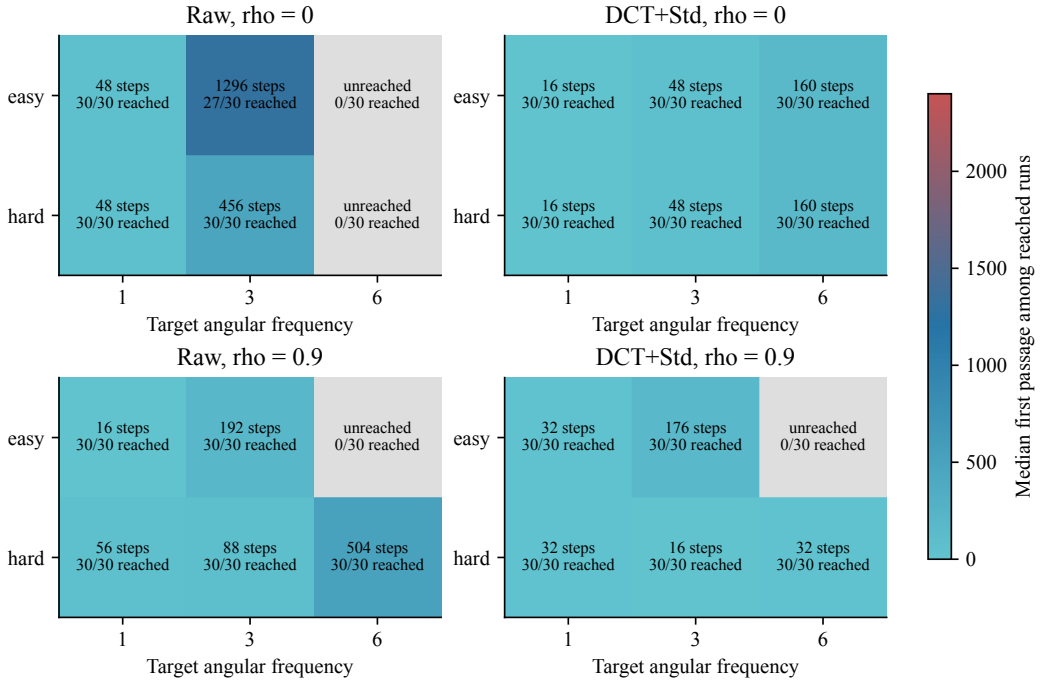


Figure 8: Learning-speed comparison with fixed physical target frequency. Definitions of threshold, medians, counts, and gray cells follow Figure 6. Keeping the frequency vector fixed across correlations distinguishes this experiment from training-standardized projections.

E Numerical Characterization of the Hessian Approximation

This experiment directly examines the B-spline Hessian underlying Proposition 1. It complements the forecasting experiments with exact matrix calculations, providing a quantitative characterization of the covariance-transfer approximation across correlation levels, spline resolutions, and sample sizes.

E.1 Fixed-basis model and matrix construction

We use a single-layer model with fixed cubic B-spline basis functions and trainable linear coefficients only. With design matrix Φ and loss $\|\Phi\boldsymbol{\theta} - \mathbf{y}\|_2^2/(2n)$, the empirical Hessian is exactly

$$M_{\text{exact}} = \frac{1}{n}\Phi^\top\Phi, \qquad M_{\text{approx}} = J_p \otimes D + R \otimes (C - D). \tag{119}$$

No optimization or labels are required for this calculation. The experiment uses B-splines directly, matching the theoretical model rather than the RBF parameterization of FastKAN.

Independent Gaussian lag vectors are sampled with population correlation $R_{ij}^{\text{pop}} = \rho^{|i-j|}$. The full grid is $p \in \{3, 6\}$, $G \in \{4, 8, 16\}$, $\rho \in \{0, 0.3, 0.6, 0.9\}$, and $n \in \{10^4, 10^5\}$, with five Monte Carlo seeds: 240 matrix settings in total. Here, G denotes the number of equal-width intervals on $[-4, 4]$, giving $G + 3$ cubic B-spline basis functions per coordinate with an open clamped knot sequence. Values outside this interval are clipped, affecting at most 0.0133% of sampled coordinates. We use the actual post-clipping empirical correlation matrix R and pool equal-marginal coordinate observations to estimate C and D. All calculations use float64.

For each $(p, G, \rho, \text{seed})$, an independent set of 10^6 vectors supplies a reference estimate, giving 120 reference calculations. The 10^4-sample estimate is a prefix of the corresponding 10^5-sample stream; these sample sizes are compared as paired estimates. The independent reference separates persistent approximation error from finite-sample variation.

We explicitly remove the $p - 1$ constant-reparameterization directions using an orthonormal basis U for their complement. The reduced matrices $U^\top MU$ have dimension $1+p(G+2)$. Effective condition numbers use positive eigenvalues exceeding $\tau\lambda_{\max}$, with $\tau \in \{10^{-10}, 10^{-8}, 10^{-6}\}$; tabulated values use $\tau = 10^{-8}$. Additional numerical rank deficiency is recorded separately from the removed constant directions.

E.2 Approximation accuracy across spline resolutions

Figure 9 presents the complete error grid, measured by

$$\epsilon_F = \frac{\|M_{\text{exact}} - M_{\text{approx}}\|_F}{\|M_{\text{exact}}\|_F}, \qquad \epsilon_2 = \frac{\|M_{\text{exact}} - M_{\text{approx}}\|_2}{\|M_{\text{exact}}\|_2}. \tag{120}$$

The coarse-grid model provides a particularly accurate matrix approximation: at $G = 4$ and $n = 10^5$, the mean relative Frobenius error remains below 1.9% for every tested p and ρ. In the independent-input case, increasing the sample size reduces the error across all grid sizes; the independent million-sample estimates have errors between approximately 0.06% and 0.29%. These results numerically recover the independent-input structure and identify a regime in which the leading-order closure closely matches the full Hessian.

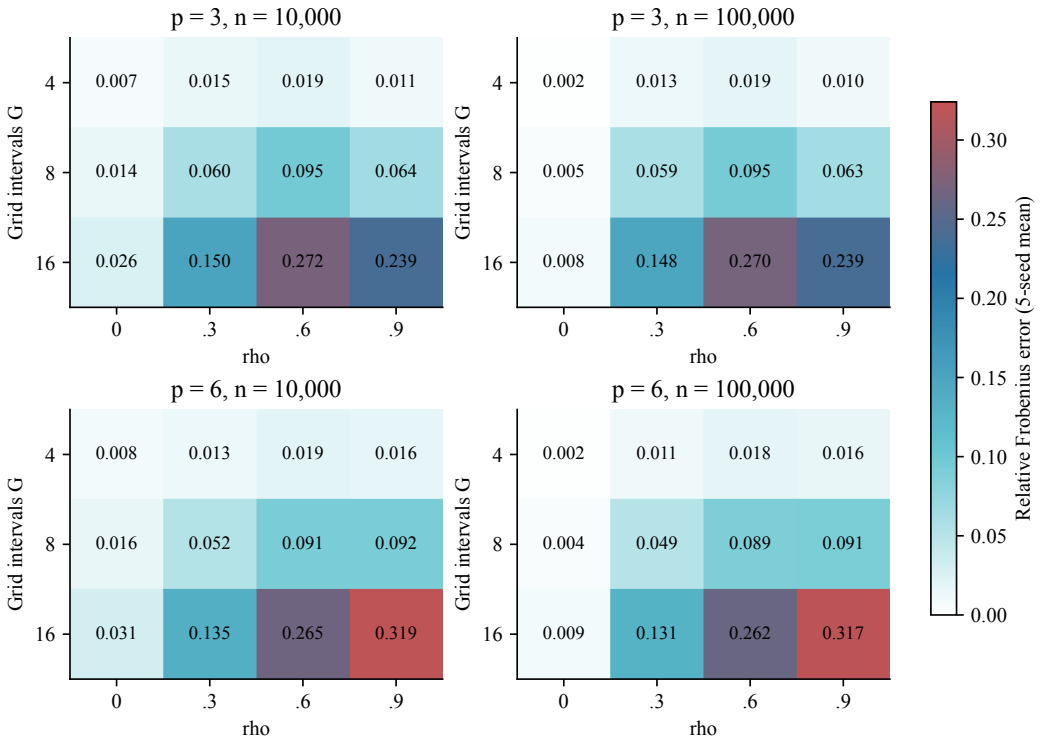


Figure 9: Relative Frobenius error of the covariance-transfer Hessian approximation over the complete $p \times G \times \rho \times n$ grid. Each entry is the mean of five Monte Carlo seeds. The paired sample-size comparison distinguishes sampling variation from the correlation- and resolution-dependent approximation residual.

Table 4 provides all $n = 10^5$ settings alongside the independent million-sample error. At finer resolutions, the closure retains a systematic residual under correlated inputs. For example, $p = 6$, $G = 16$, and $\rho = 0.9$ give $\epsilon_F = 0.3171$ and $\epsilon_2 = 0.2143$; the independent reference gives $\epsilon_F = 0.3169$. The persistence of this discrepancy at the larger sample size identifies it as an approximation effect. Error also depends on grid resolution and need not increase monotonically with ρ for every fixed grid.

Table 4: Complete Hessian summary at $n = 10^5$, averaged over five seeds. The independent-reference column measures exact versus approximate matrices within a separate 10^6-sample estimate. Condition numbers use the reduced matrices and relative eigenvalue threshold 10^{-8}; both condition-number columns are displayed in units of 10^3.

			Relative Frobenius error		Condition number / 10^3	
p	G	ρ	$n = 10^5$	Reference 10^6	Exact	Approximate
3	4	0	0.0023	0.0006	9.1	7.8
3	4	0.3	0.0134	0.0133	8.8	11.6
3	4	0.6	0.0189	0.0188	9.4	23.5
3	4	0.9	0.0103	0.0102	24.3	111.4
3	8	0	0.0046	0.0013	23.7	15.0
3	8	0.3	0.0588	0.0589	23.2	23.8
3	8	0.6	0.0949	0.0951	30.2	47.2
3	8	0.9	0.0632	0.0632	40.8	242.9
3	16	0	0.0084	0.0025	43.8	22.5
3	16	0.3	0.1479	0.1482	49.9	35.8
3	16	0.6	0.2704	0.2710	232.3	71.6
3	16	0.9	0.2391	0.2393	45.4	358.6
6	4	0	0.0021	0.0007	18.5	15.5
6	4	0.3	0.0110	0.0109	17.8	24.8
6	4	0.6	0.0179	0.0178	19.0	53.4
6	4	0.9	0.0158	0.0157	55.0	272.8
6	8	0	0.0044	0.0015	48.5	28.7
6	8	0.3	0.0494	0.0493	46.1	44.8
6	8	0.6	0.0893	0.0893	54.4	101.9
6	8	0.9	0.0908	0.0907	86.3	592.9
6	16	0	0.0089	0.0029	65.5	41.5
6	16	0.3	0.1311	0.1310	73.1	69.0
6	16	0.6	0.2618	0.2618	115.0	152.2
6	16	0.9	0.3171	0.3169	143.0	843.8

E.3 Spectral structure and condition-number bounds

Figure 10 compares the full reduced spectra for the fine-grid setting $p = 6$, $G = 16$, at $\rho = 0$, 0.6, and 0.9. For this setting, the mean exact effective condition number increases from 6.55×10^4 at $\rho = 0$ to 1.43×10^5 at $\rho = 0.9$, demonstrating that autocorrelation affects the optimization spectrum even when the Hessian is computed directly from the spline design matrix. The approximate condition number changes from 4.15×10^4 to 8.44×10^5, giving a stronger amplification and a visible discrepancy in the spectral tail.

The approximate matrices satisfy the checked condition-number interval in Eq. (18) for all 240 settings. For the exact matrices, 116 of the 240 threshold-based effective condition numbers fall within that interval. The interval therefore characterizes the leading-order model, while sharper bounds for the exact Hessian require the measured covariance residual. At $p = 6$, $G = 16$, $\rho = 0.9$, and $n = 10^5$, the mean ratio $\|M_{\text{exact}} - M_{\text{approx}}\|_2 / \lambda^+_{\min}(M_{\text{exact}})$ is approximately 3.06×10^4, explaining why matrix-level agreement alone does not ensure agreement of the smallest positive eigenvalues.

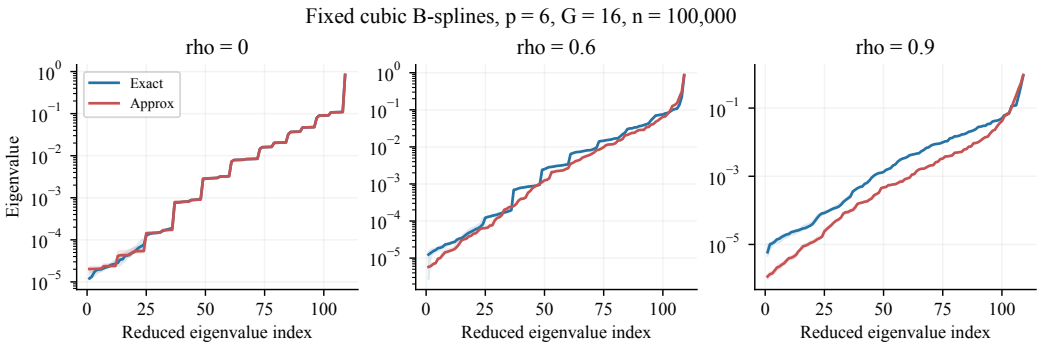


Figure 10: Exact and approximate reduced Hessian spectra for $p = 6$, $G = 16$, and $n = 10^5$. The $p - 1$ constant-reparameterization directions have been removed. Lines show the median over five seeds, and shading shows the seedwise minimum–maximum range. All reduced eigenvalues are displayed, including the spectral tail.

At $n = 10^4$, the reduced exact matrices exhibit up to five additional numerical rank deficiencies at threshold 10^{-8}; none remains at $n = 10^5$ at this threshold. Figure 11 further localizes the closure residual by lag separation, using absolute covariance error so that the independent-input case remains interpretable when the true cross-covariance is zero.

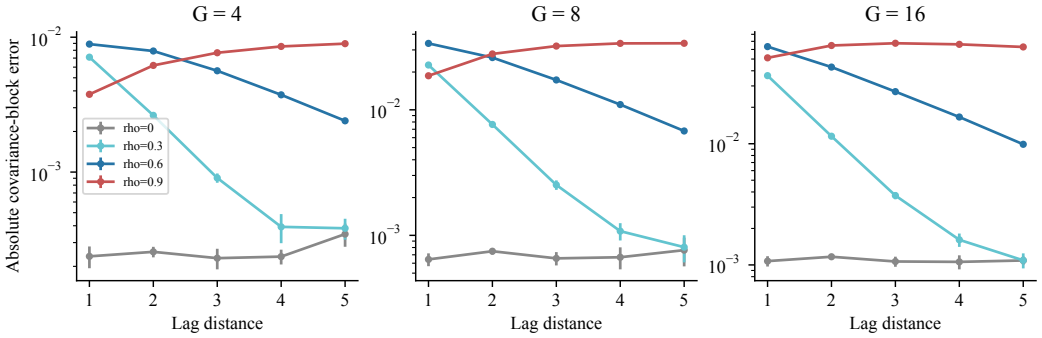


Figure 11: Absolute B-spline cross-covariance approximation error by lag distance for $p = 6$ and $n = 10^5$. Within each seed, errors are averaged over coordinate pairs at the specified distance. Points show five-seed means and error bars show seed standard deviations. Panels compare the three spline resolutions.

Together, these calculations give the theoretical analysis a concrete numerical foundation: the coarse-grid closure is accurate at the matrix level, the exact spline Hessian responds to input correlation, and the approximation residual is explicitly measurable. Incorporating its higher-order covariance structure into sharper spectral-tail bounds is a focused direction for future work.

Implementation verification. The saved experiment checks confirm agreement between the cubic B-spline implementation and SciPy's design matrix, the analytical and automatic-differentiation Hessians, and the explicit constant-nullspace construction. For E2, checks cover independent versus batched FastKAN forward passes and Adam updates, paired labels, complete per-epoch records, prediction-based MSE recomputation, and the known-answer component metrics. The full experiment grid completed without numerical failures; unreached learning thresholds remain explicitly recorded.

F Detailed Protocol for the Synthetic Forecasting Experiments

We conduct controlled experiments on synthetic autoregressive time series to study the influence of input autocorrelation on the spectral behavior of KANs. The univariate time series is generated by an AR(N) process:

$$x_t = \rho_1 x_{t-1} + \rho_2 \sum_{i=2}^{N} x_{t-i} + \sqrt{1 - \rho_1^2 - (N-1)\rho_2^2}\,\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,1), \tag{121}$$

where ρ_1 controls the main autocorrelation strength. The forecasting input is constructed as

$$\mathbf{X}_t = [x_t, x_{t-1}, \ldots, x_{t-p+1}]^\top, \qquad p = 3N. \tag{122}$$

All inputs are standardized coordinate-wise using statistics computed from the training set.

To construct target functions with explicit frequency components, we use two projection directions:

$$\mathbf{w}_{\text{easy}} = \mathbf{1}_N \otimes [1,1,1]^\top, \qquad \mathbf{w}_{\text{hard}} = \mathbf{1}_N \otimes [1,-2,1]^\top, \tag{123}$$

where $\otimes$ denotes the Kronecker product, so that the basic patterns $[1,1,1]$ and $[1,-2,1]$ are repeated N times, respectively, and both vectors have dimension $p = 3N$. The vector $\mathbf{w}_{\text{easy}}$ acts as a local averaging direction and mainly captures smooth, low-frequency variations along the lag dimension. In contrast, $\mathbf{w}_{\text{hard}}$ is a repeated second-difference pattern, which suppresses smooth trends and emphasizes oscillatory lag structures. Therefore, these two directions allow us to separate easy low-frequency components from harder high-frequency components.

Let

$$V_{\text{easy},t} = \mathbf{w}_{\text{easy}}^\top \widetilde{\mathbf{X}}_t, \qquad V_{\text{hard},t} = \mathbf{w}_{\text{hard}}^\top \widetilde{\mathbf{X}}_t, \tag{124}$$

where $\widetilde{\mathbf{X}}_t$ denotes the standardized input. Both projected variables are further normalized to zero mean and unit variance. The target is defined as

$$y_{t+1} = \sin(\omega_{\text{low}} V_{\text{easy},t}) + \sin(\omega_{\text{mid}} V_{\text{hard},t}) + \sin(\omega_{\text{high}} V_{\text{hard},t}) + \eta_t, \tag{125}$$

where $\eta_t \sim \mathcal{N}(0, 0.05^2)$. The detailed settings are summarized in Table 5.

Table 5: Experimental settings for the synthetic spectral-bias experiments.

Item	**Setting**
Data-generating process	AR(N) process in (121).
Autocorrelation parameter	$\rho_1 \in \{0.1, 0.2, \ldots, 0.8\}$.
AR configurations	$(N, p, \rho_2) \in \{(1,3,0), (2,6,0.1), (5,15,0.01)\}$.
Sequence length	5000 samples.
Train/test split	80% / 20%, split chronologically.
Number of runs	100 independent runs for each ρ_1.
Target frequencies	$\omega_{\text{low}} = 1$, $\omega_{\text{mid}} = 3$, $\omega_{\text{high}} = 6$.
Observation noise	$\eta_t \sim \mathcal{N}(0, 0.05^2)$.
Model	FastKAN with width $= [p, 50, 1]$, grid $= 8$.
Optimizer	Adam with learning rate 0.005.
Batch size / epochs	256 / 150.
KAN input	Standardized lagged vector $\widetilde{\mathbf{X}}_t$.
DCT-KAN input	Orthonormal DCT of $\widetilde{\mathbf{X}}_t$ along the lag dimension, followed by coordinate-wise standardization using training-set statistics.
Evaluation metrics	Test MSE and component-wise recovery errors for low-, middle-, and high-frequency components.

For component-wise evaluation, let $\hat{y}$ denote the prediction and C_q the known synthetic component, $q \in \{\text{low}, \text{mid}, \text{high}\}$. The recovered amplitude and component error are

$$\hat{a}_q = \text{clip}\left(\frac{\text{Cov}(\hat{y}, C_q)}{\text{Var}(C_q) + 10^{-8}}, 0, 1\right), \qquad E_q = (1 - \hat{a}_q)^2. \tag{126}$$

A smaller E_q indicates better recovery of the corresponding frequency component.