Mechanistic Evidence for Spectral Structures in Prior-Data Fitted Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Prior-Data Fitted Networks (PFNs) enable amortized Bayesian inference in a single forward pass, yet their internal representations remain opaque. It is unknown whether PFNs encode identifiable Bayesian structure or merely memorize input-output mappings. We provide mechanistic evidence that PFNs learn structured spectral representations and that these can be extracted as explicit kernels. First, probing experiments across three architectures, including the publicly released TabPFN, show that spectral information is linearly decodable from the latent attention score and organized along a dominant principal axis. Activation patching and targeted subspace interventions establish that this information is causally used for prediction and concentrated in a low-dimensional subspace, with spectral directions an order of magnitude more effective than random ones. Crucially, these properties hold on TabPFN with both synthetic out-of-distribution inputs and real-world time series (Airline Passengers, Milk Production), indicating they are emergent features of PFN-style amortization over continuous regression tasks rather than artifacts of training prior. Second, we introduce a Filter Bank Decoder that maps frozen PFN latents to explicit spectral densities, reconstructing stationary kernels via Bochner's theorem. The resulting kernels support GP regression competitive with iterative baselines while requiring only a single forward pass, demonstrating that PFN priors are not merely implicit but are explicitly recoverable as portable Bayesian objects.

1 Introduction

Bayesian Inference methods like Gaussian processes (GPs) provide a principled way to learn and reason under uncertainty. The explicit representation in terms of kernel is valuable wherever understanding the data-generating process is as important as making accurate predictions. However, inference with expressive kernels is computationally expensive, and restricted to fixed kernel families, limiting scalability and repeated use [Rasmussen and Williams, 2006].

Prior-Data Fitted Networks (PFNs) offer a resolution to this computational bottleneck [Müller et al., 2022b]. A PFN is trained on a distribution of synthetic tasks sampled from a prior, learning to map context data directly to posterior predictive distributions (PPD) in a single forward pass. This enables efficient inference across new datasets drawn from the same prior family.

This efficiency comes at the cost of **structural opacity**. In a classical GP, the kernel is an explicit, interpretable, portable object that can be inspected, composed [Duvenaud et al., 2013], and transferred to new tasks. In a PFN, the corresponding structure is considered to be implicitly absorbed into the network's weights and activations. The model produces approximate PPD, but it is not known if the hidden Bayesian object has any interpretable correlation with the input. It is also unknown whether any such object, like a kernel matrix, can be extracted from frozen PFN and provides comparative downstream performance as other kernel design methods like [Wilson et al., 2015, Duvenaud et al., 2013, Lloyd et al., 2014, Wilson and Adams, 2013].

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

Recently, [Müller et al., 2025] surveyed the state of PFN research and identified several open challenges that must be addressed for PFNs to serve as general-purpose Bayesian tools. Among these, two stand out as particularly fundamental: **1) the lack of mechanistic transparency**: we do not understand *where* or *how* Bayesian structure is represented inside the network, nor whether it plays a causal role in prediction and **2) the absence of explicit, portable representations** of the latent priors learned during amortized inference. This work addresses both gaps by asking *two complementary questions*: **(a)** *Do PFNs encode spectral information in a structured and interpretable form, and is this information causally used in prediction?* and **(b)** *Can this structure be extracted as explicit kernel representations?*

Answering these questions requires mechanistic analysis tools that go beyond current practice. Mechanistic analysis methods have been predominantly developed for and validated on large language models [Bills et al., 2023, Bricken et al., 2023, Belinkov, 2022], with non-language foundation models comparatively underexplored [El et al., 2025]. Even within that setting, analyses are typically descriptive as they reveal what information is present but do not extract representations that can be reused for downstream tasks. This gap is particularly relevant for models such as PFNs, as highlighted recently in Müller et al. [2025]. We address this gap by showing that mechanistic structure in PFNs can not only be identified but also *operationalized* for better PFN design and downstream tasks.

In this paper, to answer the aforementioned question **(a)**, we provide two complementary lines of evidence. First, probing experiments demonstrate that spectral information is *linearly decodable* and cleanly organized within the latent attention score. Second, activation patching and targeted subspace interventions establish that this information is *causally used by the network* and compressed into a low-dimensional subspace. Crucially, we show that these phenomena are not specific to a single architecture: spectral organization emerges in TabPFN [Hollmann et al., 2025], a standard PFN with joint attention trained on tabular data. A GP-specialized model, DVA-PFN [Sharma et al., 2025], exhibits significantly clearer spectral structure when trained directly on spectral priors. Importantly, these properties hold across 1D sinusoidal probing inputs, 5D RBF and Matérn GP inputs, and tabular inputs, confirming the generality of the analysis. We further validate these causal findings on *real-world time series* converted to tabular regression via lag embedding.

For question **(b)**, building on the mechanistic foundation, we introduce a *Filter Bank Decoder* that maps frozen PFN representations to explicit spectral density estimates. The resulting kernels support GP regression competitive with iterative baselines while requiring only a single forward pass, and enable downstream tasks with performance competitive with kernels obtained via iterative and amortized kernel discovery methods like [Wilson et al., 2015, Bitzer et al., 2023, Tancik et al., 2020]. In summary, our contributions are:

- We provide first systematic mechanistic study of spectral structure in PFNs which is linearly decodable and causally relevant latent subspace. Also we show that it is a low-dimensional subspace, consistent across architectures, and validated on real-world observational data.
- We introduce a Filter Bank Decoder that extracts explicit, portable kernels from frozen PFN representations, enabling competitive GP regression and downstream Bayesian tasks without test-time optimization.

Together, our results show that PFNs not only approximate Bayesian inference but also learn structured and extractable representations of prior knowledge. We also provide some evidence in directions to improve PFN design based on these mechanistic analyses.

2 Background

2.1 Prior-Data Fitted Networks

Recently, [Müller et al., 2022b] enabled amortized Bayesian inference by training a neural set-predictor on a vast distribution of synthetic datasets. Formally, given a prior $p(\mathcal{D})$ over supervised learning tasks, we sample datasets $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N} \sim p(\mathcal{D})$. Each dataset is partitioned into a context set $\mathcal{D}_{\text{ctx}} = (X_{\text{ctx}}, Y_{\text{ctx}})$ containing observed pairs, and a query set $\mathcal{D}_{\text{q}} = (X_{\text{q}}, Y_{\text{q}})$ containing targets to predict. The model parameters θ are optimized to minimize the expected Negative Log-Likelihood loss. Further, there has been growing interest in training these PFNs on different priors for specific tasks Müller et al. [2025]. Among them, *tabular prior* based TabPFN Grinsztajn et al. [2025] has gained significant traction due to its ability to work with real-world tabular data, while only trained

on synthetic dataset. Thus, solving the data availability problem for various applications Hollmann et al. [2025], Liu and Ye [2025], Feuer et al. [2024]. We select standard vanilla attention (VA) PFN Müller et al. [2022b] and TabPFN (Version 2.5) Grinsztajn et al. [2025] along with Decoupled-Value Attention (DVA) PFN presented in Sharma et al. [2025]. Together these three provide spectrum of attention mechanisms ranging from alternating row/column attention (TabPFN), to joint input-output attention (VA-PFN) and decoupled input-output attention forcing localization (DVA-PFN).

2.2 Mechanistic Analysis Methods

Mechanistic analysis aims to reverse-engineer the internal computations of neural networks by identifying interpretable features, and algorithms learned during training [Olah et al., 2020, Elhage et al., 2021]. Techniques like activation patching and automated circuit discovery have enabled fine-grained analysis of individual components contribution [Conmy et al., 2023]. A complementary approach is the use of *probing classifiers* [Alain and Bengio, 2016, Belinkov, 2022]. However, as highlighted in introduction, mechanistic analysis works are limited to language models mainly.

Probing classifiers. Probing classifiers are lightweight models trained to predict *properties of interest* from frozen intermediate representations of a neural network [Alain and Bengio, 2016] i.e, trained to test what information models encode. A linear probe lower bounds what is linearly accessible; a higher-capacity probe can recover nonlinearly entangled information, but risks learning the task itself [Hewitt and Liang, 2019, Belinkov, 2022]. Further, probing is purely correlational: high probe accuracy demonstrates that information is *present* in a representation, but not that the network *uses* it during inference [Belinkov, 2022]. This limitation motivates the causal methods.

Activation patching. Activation patching [Meng et al., 2022, Vig et al., 2020] tests causality by replacing a representation at layer ℓ of input A with that of input B and measuring how far the output moves toward B's prediction (the *causal effect)*. This technique is also referred to as causal tracing Meng et al. [2022] or interchange intervention Geiger et al. [2021], has been used extensively in language models to localize factual knowledge and identify task-specific circuits Conmy et al. [2023], Wang et al. [2023] Patching different sites disentangles the causal roles of sub-modules.

Targeted subspace interventions. Full activation patching replaces an entire representation vector, leaving open the question that whether the causally relevant information occupies the full ambient dimensionality or is concentrated in a compact subspace. Targeted subspace patching [Geiger et al., 2024] replaces only the top-k principal components ranked by correlation with the property of interest, thus resulting *dose–response curve* bounds both dimensionality and geometry of causal subspace.

2.3 Spectral Representation of Stationary Kernels

For a stationary covariance kernel $k(\tau)$, Bochner's theorem [Bochner, 1959] establishes a one-to-one correspondence between the kernel and a non-negative spectral density $S(\omega)$ via the Fourier transform: $k(\tau) = \int_{\mathbb{R}} S(\omega)\, e^{2\pi i \omega \tau}\, d\omega$, $S(\omega) \geq 0$. This duality means that specifying a kernel is equivalent to specifying a spectral density: smoothness, periodicity, and long-range correlation structure are all encoded in the shape of $S(\omega)$. Building on this correspondence, Wilson and Adams [2013] introduced the *Spectral Mixture* (SM) kernel, which models the spectral density as a mixture of Gaussians: $S(\omega) = \sum_{q=1}^{Q} w_q \mathcal{N}\big(\omega \mid \mu_q,\, \sigma_q^2\big)$, $w_q > 0$, where ,each component is characterized by a center frequency μ_q, a bandwidth σ_q, and a mixture weight w_q. Substituting the density into Bochner theorem result yields the closed-form kernel

$$k(\tau) = \sum_{q=1}^{Q} w_q \exp\!\big(-2\pi^2 \sigma_q^2 \tau^2\big) \cos(2\pi \mu_q \tau)\,. \tag{1}$$

The SM kernel is a universal approximator in the space of stationary kernels [Wilson and Adams, 2013]: any continuous stationary kernel on a compact domain can be approximated arbitrarily well by choosing a sufficient number of mixture components Q. This expressiveness makes the spectral density a natural target for kernel discovery—recovering $\{(\mu_q, \sigma_q, w_q)\}_{q=1}^{Q}$ from data is equivalent to recovering the full covariance structure of the underlying process. Importantly, there has been considerable work in kernel design and discovery Wilson and Adams [2013], Wilson et al. [2015], Tancik et al. [2020] most of which requires per-sample optimization to find kernel matrix or hyperparameters except notable works like [Bitzer et al., 2023].

3 Locating Spectral Structure in PFN Representations

We first ask how a trained PFN organizes spectral information about input. We answer this with a ladder of probes of increasing capacity, following [Alain and Bengio, 2016] [Belinkov, 2022]: each rung adds parameters and tests a stricter notion of accessibility. Throughout, we compare the three architectures Section 2.1, which vary in attention design, training prior, and scale. VA-PFN and DVA-PFN are trained by us on spectral mixture priors; TabPFN is a publicly released frozen model [Grinsztajn et al., 2025] trained on structural causal models not on spectral kernels, with a tabular task format. Agreement across all three is strong evidence that our findings reflect a general property of PFNs over continuous inputs rather than an artifact of an architecture or a prior.

3.1 Parameter-Free Probing: Frequency-Driven Geometry of $\bar{H}$

The most conservative probe has zero trainable parameters. We ask a simple question: *when the generating frequency f of an input sinusoid changes, does the latent $\bar{H}$ change in a correspondingly structured way?* We generate 500 sinusoids $y(t) = \sin(2\pi f_i t + \phi)$ with $f_i \sim \mathcal{U}[0.5, 5.0]$ Hz and random phase $\phi \sim \mathcal{U}[0, 2\pi]$, pass each through the frozen PFN, and mean-pool the final-layer attention score over positions (t values) to obtain $\bar{H} \in \mathbb{R}^d$. Here, d is a PFN hyperparameter reflecting dimension into which input is projected while $H = QK^T/\sqrt{d}$ are attention scores.

First, we attempt to answer *is the global geometry of $\bar{H}$ governed by f?* We calculate the Pearson correlation between two vectors of pairwise distances as $\rho_\Delta = \mathrm{Corr}(\{\Delta f_{ij}\}, \{\Delta H_{ij}\})$, which measure change in latent embedding ($\Delta H_{ij} = \|\bar{H}_i - \bar{H}_j\|_2$) with change in input frequency ($\Delta f_{ij} = |f_i - f_j|$). This measures whether frequency-close signals are embedding-close. Second, we ask *if the frequency sensitivity is dominant in a particular direction in $\bar{H}$?*. For this, we report the Pearson correlation $|r|_{\mathrm{PC}_0}$ between the generating frequency and $\bar{H}$ projected onto its first principal component in the original d-dimensional space. Let $\mathrm{PC}_0 = \arg\max_{\|v\|=1} v^\top \Sigma v$ with $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\bar{H}_i - \mu)(\bar{H}_i - \mu)^\top$. Then we calculate $|r|_{\mathrm{PC}_0} = \left|\mathrm{Corr}\left(\{z_i\}_{i=1}^N, \{f_i\}_{i=1}^N\right)\right|$ with $z_i = \mathrm{PC}_0^\top(\bar{H}_i - \mu)$. This isolates the dominant mode of variation rather than the full geometry.

Table 1: Parameter-free frequency alignment metrics ρ_Δ and $|r|_{\mathrm{PC}_0}$ (mean $\pm$ std over 5 seeds).

Metric	VA-PFN	TabPFN	DVA-PFN
$\boldsymbol{\rho_\Delta}$	0.615 ± 0.038	0.765 ± 0.021	0.852 ± 0.015
$\boldsymbol{\lvert r\rvert_{\mathrm{PC}_0}}$	0.812 ± 0.026	0.882 ± 0.010	0.981 ± 0.006

Table 1 reports both numbers for all three PFNs. Most significantly, in TabPFN, which was trained on synthetic tabular priors, $\bar{H}$ tracks frequency at $\rho_\Delta = 0.76$ and $|r|_{\mathrm{PC}_0} = 0.88$ shows that frequency sensitivity in a PFN's latent does not require training on spectral kernel priors, it transfers to inputs well outside the training distribution. This suggests that PFN-style amortization over continuous regression inputs reliably gives rise to this representation across the architectures tested. Also, the ordering VA $<$ TabPFN $<$ DVA directly tracks the degree to which the attention head carries input information only separates the value stream from the query–key pair, as explained next.

Where Does Spectral Information Live? Repeating the parameter-free measurement on the value stream $\bar{V}$ gives a sharp architectural split: $\rho_V = 0.19$ (DVA), 0.53 (VA), 0.79 (TabPFN), with the t-SNE visualizations in Figures 4a and 4b (Appendix B) confirming the same ordering. This matches how each mechanism handles the value stream: DVA-PFN routes y-information to $\bar{V}$ alone and confines frequency to $\bar{H}$; VA lets the joint (x, y) embedding mix into all three projections, so frequency leaks into $\bar{V}$; and TabPFN's alternating row/column attention repeatedly updates value representations alongside feature context, producing the strongest leakage. Thus, we read the $\bar{H}$–$\bar{V}$ split as architectural evidence that attention design governs the *localization* only not the *existence*.

3.2 How Accessible Is the Spectral Signal in $\bar{H}$?

Geometric alignment (Sec. 3.1) tells us $\bar{H}$ is *organized* by frequency, but not whether spectral quantities can be *read out* in a simple form. Following the probe ladder of Alain and Bengio [2016] and Hewitt and Liang [2019], we train two probes of increasing capacity on the same frozen $\bar{H}$: a linear probe (lower bound on accessibility) and a nonlinear MLP probe (upper bound). Our hypothesis

is that if these two agree with high R^2 value, it implies MLP complexity is not required and the target is encoded in an approximately linearly separable form. (Appendix A.3 details of probes).

Control: random weights. To confirm that linear accessibility reflects learned structure rather than trivial input geometry [Hewitt and Liang, 2019], we repeat the probing experiment on a randomly initialized (untrained) VA-PFN as a control. The MLP probe still succeeds ($R^2 \geq 0.99$), but the linear probe collapses to $R^2 = 0.18$ (frequency) and 0.64 (weight), confirming that training specifically organizes $\bar{H}$ into a linearly separable form [Belinkov, 2022]. See Figure 7 in Appendix.

A linear read-out is enough. Table 2 shows that a linear probe on $\bar{H}$ recovers both frequency and weight with $R^2 \geq 0.93$ across all three PFNs. The ordering in Table 1 repeats under probing: DVA $>$ TabPFN $>$ VA. In particular, an off the shelf TabPFN, supports $R^2 = 0.993$ for frequency recovery, the clearest single-task evidence that spectral organization is a recurring property of the amortized Bayesian predictors tested here. Further, a we train a higher-capacity MLP probe, which can recover information that is present but nonlinearly entangled. The right-hand columns of Table 2 report the gap $\Delta = R^2_{\text{MLP}} - R^2_{\text{linear}}$. The gap is within ± 0.02 for every task and every architecture, and is slightly *negative* on VA-PFN, consistent with mild over-fitting of the larger probe. This low gap along with the $R^2 \to 1$ for both probes show that PFNs internally solved the representation-learning problem for the scalar targets we probe and frequency and weights are exposed as approximately affine functions of the coordinates of $\bar{H}$. This rules out the hypothesis that PFNs store spectral information in a form requiring nonlinear post-processing. See Figure 9-10 for alignment with true frequency.

Table 2: Linear vs. nonlinear probing on $\bar{H}$. For single-component signals, both quantities are linearly decodable across all three architectures, and added MLP capacity gives no consistent improvement.

		VA-PFN		TabPFN		DVA-PFN	
Task	Probe	R^2	Δ	R^2	Δ	R^2	Δ
Frequency	Linear	0.967	–	0.993	–	0.998	–
	MLP	0.946	-0.021	0.991	-0.002	1.000	$+0.002$
Weight	Linear	0.934	–	0.982	–	0.997	–
	MLP	0.913	-0.021	0.976	-0.006	0.999	$+0.002$

Learned rectification enables mean-pooling Since $\int \sin(\omega t)\, dt \to 0$, recovering f at $R^2 \to 1$ from mean-pooled $\bar{H}$ (Table 2) implies that the PFN's MLP layers apply a rectifying nonlinearity before aggregation, a learned analogue of a classical periodogram's square-and-average step. This accessibility weakens for multi-component signals: recovering all four parameters (f_1, f_2, a_1, a_2) drops to $R^2 = 0.50$ (Table 5; Figure 8; Table 4 in Appendix), a failure of uniform aggregation we address with multi-query attention pooling in Sec. 5.

3.3 Non-sinusoidal, Higher-dimensional Inputs

Beyond 1D sinusoids, we repeat the parameter-free probing analysis on 5D functions drawn from GP with RBF and Matérn-3/2 kernels. For each kernel family we sample 500 functions with lengthscales (ℓ) log-uniformly distributed over $[0.05,\ 10]$, pass them through frozen TabPFN, and compute the same two metrics against the characteristic spectral frequency $f_{\text{char}} = 1/(2\pi\ell)$ (or $\sqrt{3/2}/(2\pi\ell)$ for Matérn). We report $\rho_\Delta = 0.815$ and $|r|_{\text{PC0}} = 0.900$ for RBF, and $\rho_\Delta = 0.861$ and $|r|_{\text{PC0}} = 0.928$ for Matérn-3/2 (see Table 6 in Appendix). These values are comparable to those obtained on 1D sinusoids in Table 1. This along with t-SNE plots for both inputs (Figure 5 in Appendix), confirms that the spectral organization of $\bar{H}$ generalizes beyond 1D sinusoidal signals, further supporting our claim that structured spectral encoding is a general property of PFNs over continuous inputs.

Layer-Wise Spectral Refinement. We next ask how the spectral representation evolves across depth by extracting $\bar{H}$ at every layer and computing the correlation ρ_Δ. Both DVA and TabPFN exhibit a characteristic *rise–plateau–decline trajectory* (Figure 6, Appendix). The early saturation confirms that a single cross-attention step suffices to fuse positional and value information into a spectrally meaningful latent, while the late-stage dip is consistent with the final layers reallocating capacity from geometric separation toward formatting the calibrated posterior predictive distribution.

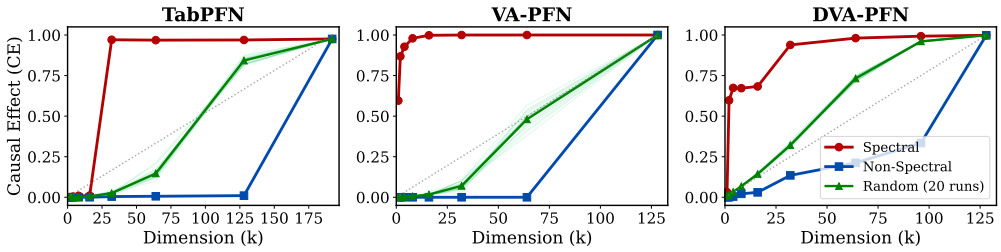


Figure 1: Causal dose–response under targeted subspace patching across architectures. Each panel sweeps patched directions k from 1 to d and reports the causal effect (CE) for spectral (red), non-spectral (blue), or random (green, 20 trials) subspaces.

4 Is the Encoding Causal?

Previous section observations are correlational as a high probe R^2 shows that information is *present* in the representation, not that the network *uses* it during inference [Belinkov, 2022]. We close this gap with two *interventional experiments* on the frozen PFNs: **(a) Activation patching** [Meng et al., 2022, Vig et al., 2020] to show that $\bar{H}$ is a causal carrier of spectral identity. **b Targeted subspace patching** to see if causally-active information occupies the full latent or a compact subspace [Geiger et al., 2021]. We than perform patching using tabular data for in-distribution assessment of TabPFN.

4.1 Activation Patching

We run two sinusoidal signals A and B (frequency f_A and f_B), through the frozen DVA-PFN and cache intermediate representations at every layer. We then *patch* (replace) the representation of signal A with that of signal B at layer ℓ and continue the forward pass. The **Causal Effect** (CE, (2)) is the fraction by which the prediction shifts from A toward B and CE $= 1$ corresponds to a complete identity transfer. Three intervention sites isolate distinct causal pathways. The **H-patch** ($\mathbf{h}_A^{(\ell)} \leftarrow \mathbf{h}_B^{(\ell)}$) tests our central claim, that the latent attention score causally carries spectral identity. The **V-patch** ($\mathbf{v}_A \leftarrow \mathbf{v}_B$) is a positive control: V encodes only the raw function values in DVA-PFN so replacing it is equivalent to replacing the input data and CE should approach unity. The **K-patch** ($\mathbf{k}_A \leftarrow \mathbf{k}_B$) is a negative control: A and B share the same t-grid in $\sin(2\pi ft + \phi)$, so $K_A \approx K_B$ and intervention should have negligible effect. We evaluate $n = 50$ random pairs ($f \in [0.5, 5.0]$ Hz, minimum gap ≥ 1.0 Hz) across all six layers in DVA-PFN.

Further, replacing $\bar{H}$ at any layer $\ell \geq 2$ shifts the prediction completely to match the donor signal (CE $= 0.999$, $p < 10^{-133}$ against the K-patch baseline). Thus, solidifying the argument that $\bar{H}$ is causal carrier of spectral information (Table 9). Second, like Figure 6 for both DVA and TabPFN, spectral encoding emerges in the first attention step. The sharp $L_1 \to L_2$ transition (CE $0 \to 0.999$) shows that a single cross-attention layer suffices to fuse spectral information into a spectrally-meaningful $\bar{H}$, and subsequent layers maintain rather than build this representation. Third, K carries no spectral information at any layer (CE $= 0$), corroborating the attention separation in DVA.

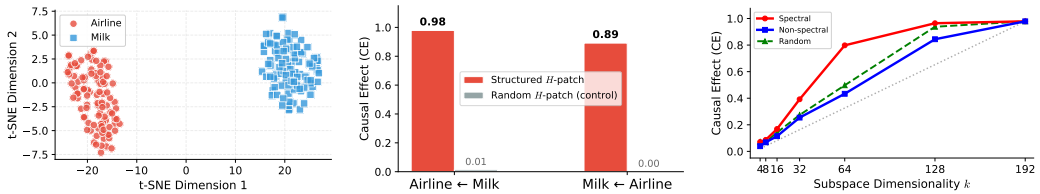
4.2 Targeted Subspace Patching

Full $\bar{H}$-patching establishes that spectral information is causally read from the latent, but leaves open a structural question: is this information distributed diffusely across the d-dimensional latent, or concentrated in a separable subspace? We localize the causally relevant directions and bound their dimensionality with a *dose–response curve* for all three PFNs. We collect $\bar{H}$ representations at Layer 2 for 2,500 signals (500 frequencies $\times$ 5 phases) and run PCA on the resulting embeddings for all PFNs. Each principal component is ranked by $|r|$, the absolute Pearson correlation between its projection and the generating frequency: the top-k components define the *spectral subspace*, the bottom-k the *non-spectral subspace*, and k random orthogonal directions a baseline. We then patch only the selected k PC dimensions of $\bar{H}_A$ with the corresponding components of $\bar{H}_B$, leaving the remaining $d - k$ dimensions intact, and sweep k from 1 to d across all three architectures.

The strongest test case here is TabPFN. Figure 1 (left) shows that patching just the top few spectral PCs of TabPFN's $\bar{H}$ shifts predictions toward the donor signal, while patching an equal number of non-spectral or random directions changes predictions meaningfully only when $k > 100$ out of 192.

Note that Table 1 already revealed a dominant spectral axis in its latent $|r|_{\mathrm{PC}_0} = 0.882$ (See Figure 12 in Appendix). This is a clear evidence that compact, causally-active spectral coding is not due to of our pretraining choices as it emerges in a model we did not train and inferred on signals well outside its training distribution i.e. an emergent property of PFNs. DVA-PFN and VA-PFN also replicate the pattern with CE $\rightarrow 1$ much faster compared to random and non-spectral baselines. See Table 10 for DVA-PFN's quantitative results with Sinusoids and Figure 15 for Dose-curves with RBF-GP for DVA-PFN and single block patching of TabPFN.

Together, these patching experiments yield two-level causal account. **(i)** $\bar{H}$ is a causal carrier of spectral information: replacing it transfers spectral identity in full (CE≈ 1). **(ii)** The information within $\bar{H}$ is compactly organized: it concentrates in a low-dimensional subspace whose dominant axis aligns with the generating frequency, and targeted intervention on this subspace is more effective, 1–2 orders of magnitude in the small-k regime, than random directions of the same size. This closes the correlational–causal loop opened in Sec. 3.2 and shows that *the spectral information probes recover from $\bar{H}$ is not a passive residue of the input but an organized representation that the network constructs in its first cross-attention step and uses for prediction.*



(a) Latent t-SNE (Block 23). (b) Full-tensor patching. (c) Subspace dose-response.

Figure 2: **Causal patching of TabPFN on real-world data. Left:** t-SNE of mean-pooled H embeddings. **Middle:** Full H-replacement and random replacement control. **Right:** Dose-Response curve replacing only top-k PCA directions (spectral), bottom-k (non-spectral) and random.

4.3 Causal Structure in Real-World Time Series

Above sections use synthetic signals only. We now validate on real time series. We apply the same patching protocol to two classic time series: **a)** monthly *Airline Passengers* (Box & Jenkins, 1949–1960) and **b)** *Milk Production* (USDA, 1962–1975). These series share seasonal periodicity ($\approx$ 12 months) but differ in trend structure, amplitude dynamics, and noise profile, making them a discriminative pair for testing whether H encodes series-specific spectral identity rather than a generic seasonal template. Each dataset is converted to a 5D tabular regression task via lag embedding ($X_t = [y_{t-1}, \ldots, y_{t-5}]$, $\hat{y} = y_t$; see Appendix C.4 for details). The latent representations at TabPFN's final block clearly separate these series in Figure 2a. Full-tensor H-replacement at TabPFN's last block achieves CE $= 0.98$ (Airline$\leftarrow$Milk) and 0.89 (reverse), while random replacement gives CE ≈ 0 in Figure 2b. Targeted subspace patching in Figure 2c reveals that the top-k PCA directions most correlated with series identity are roughly twice as causally efficient as the bottom-k (k=64 i.e. $1/3^{\text{rd}}$ of d), the spectral subspace recovers CE=0.80 versus 0.43 for the non-spectral subspace. Further, we note that the tabular patching experiment (Appendix C.3, Figure 17) demonstrates that the causal role of H extends beyond spectral identity to feature-relevance structure on TabPFN's native 8D inputs, suggesting the spectral organization we identify is one manifestation of a broader structural encoding rather than an isolated phenomenon.

5 Decoding Bayesian Structure: From Latents to Explicit Kernels

Last two sections established that $\bar{H}$ contains spectral information that is linearly accessible and causally used for prediction. Now we hypothesize that if $\bar{H}$ truly encodes input information in context of prior, kernel matrix should also be extractable. Kernel matrix is a fundamental Bayesian object providing both interpretability and downstream task capability. We pick a stationary kernel $k(\tau)$, a spectral density $S(\omega)$ as targets of extraction. Note that here the proposed decoder is not designed to compete with iterative kernel discovery methods [Lloyd et al., 2014, Duvenaud et al., 2013] (re-optimize per task) or amortized kerned discovery methods [Bitzer et al., 2023]. Rather, it serves as a constructive proof that the spectral structure identified in Sections 3–4 is rich enough to reconstruct a functional covariance which is one of the most demanding read-out of the representation.

What is recoverable from data. Two facts about identifiability shape our decoder design. *First*, from a single variance-normalized realization from spectral prior, the spectral peak *locations* and *bandwidths* of $S(\omega)$ are recoverable, but the spectral *weights* are identifiable only up to a common multiplicative constant even as $N \to \infty$. The missing global scale α admits an unbiased plug-in estimator $\hat{\alpha} = \|\mathbf{f}\|_2^2 / \mathrm{tr}(K_{pred})$, so it can be recovered analytically rather than predicted by the network. *Second*, with M independent realizations from the same prior, the full set $\{w_q, \mu_q, \sigma_q\}$ becomes identifiable. We instantiate two decoder variants matched to these regimes (single-realization and multi-realization) with frozen PFN throughout. Detailed mathematical description and proofs are given in Appendix D. Figure 3 shows design of the proposed decoder.

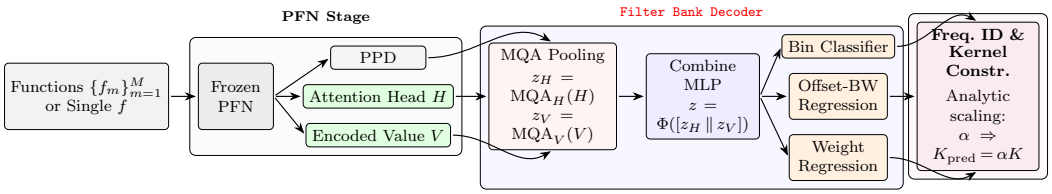


Figure 3: The proposed **Filter Bank Decoder**. Both decoders pool $\bar{H}$ and $\bar{V}$ with multi-query attention (Table 4 in Appendix shows that it is necessary once spectral complexity exceeds a single component) and predict, for each frequency bin, an activation probability, a peak offset, a bandwidth, and (multi-realization only) a weight. The predicted parameters define a spectral mixture, which is converted to a stationary kernel via Bochner's theorem (1). The PFN is never updated and the decoder is a diagnostic read-out of frozen features. Decoder details are given in Appendix E.

Predictive performance with no test-time optimization. We evaluate the decoded kernel as a plug-in covariance for GP regression on the standard kernel-cookbook benchmark (RBF, Periodic, product, Spectral Mixture with $Q \in \{1, 4\}$). Note that we evaluate not to claim state-of-the-art regression but to test whether the mechanistic structure above Sections 3-4.1 is rich enough to reconstruct a functional covariance which is one of the most demanding read-outs one could ask of any representation. **The decoder receives no information about which kernel family generated each task**: it sees only the context (X, y), runs one forward pass through the frozen PFN, and returns an explicit $k(\tau)$. Table 3 shows that, despite this fully amortized setting, the decoded kernel matches or outperforms DKL and RFF on every family and substantially so on Periodic (1.5×10^{-3} vs. 6.6×10^{-3}) and K_{RP} (2.7×10^{-3} vs. 4.9×10^{-3}), while running $\sim$250$\times$ faster, as those baselines re-optimize per task and our decoder does not. Further, the proposed decoded kernel outperforms these methods in GP-MSE, when tested on out-of-distribution kernels with varying context (Figure 16).

Table 3: GP regression MSE on kernel cookbook with 50 context points (K_{RP} denotes RBF $\times$ Periodic). The decoder serves as a diagnostic extraction, not a general predictive replacement.

True kernel	Decoder (Ours)	PFN (Ours)	Amor-struct.	DKL	RFF	Oracle GP
RBF	1.1×10^{-3}	9.7×10^{-4}	4.6×10^{-4}	8.4×10^{-4}	1.0×10^{-3}	1.7×10^{-4}
Periodic	1.5×10^{-3}	1.4×10^{-3}	1.5×10^{-3}	6.6×10^{-3}	6.4×10^{-3}	8.1×10^{-4}
K_{RP}	2.7×10^{-3}	1.1×10^{-3}	1.6×10^{-3}	4.9×10^{-3}	4.6×10^{-3}	9.5×10^{-4}
SM ($Q = 1$)	1.3×10^{-3}	4.6×10^{-4}	4.3×10^{-4}	6.4×10^{-4}	8.0×10^{-4}	2.9×10^{-4}
SM ($Q = 4$)	1.5×10^{-3}	6.2×10^{-4}	4.4×10^{-4}	1.1×10^{-3}	1.4×10^{-3}	1.9×10^{-4}
Avg. Time (s)	**0.0036**	**0.0020**	**0.0288**	**0.9180**	**0.7580**	**3.9460**

The one amortized baseline that avoids per-task optimization but receives kernel family information, Amor-struct. [Bitzer et al., 2023], does achieve lower MSE on several families, but at $8\times$ higher latency and with an architecture explicitly designed for kernel regression rather than extracted as a diagnostic from a frozen, general-purpose predictor. Critically, the decoder is not designed to compete with the PFN's own predictions, which retain access to the full latent; rather, it serves as a *constructive test* of the mechanistic findings of Secs. 3.2–4: if the spectral structure we identified in $\bar{H}$ is real and rich enough to matter, it should be possible to extract it as a functional kernel. The fact that a 9-parameter spectral-mixture object read out from frozen activations stays within a small constant factor of the PFN and beats iterative baselines that have no such structural constraint, confirms exactly this. A natural concern is that a classical FFT-based pipeline could recover $S(\omega)$ from raw (X, y) without any PFN. We test this directly (Table 14) by fitting spectral mixtures to the standard periodogram and Lomb–Scargle periodogram: both achieve very low kernel-MSE on the context

set by overfitting the discrete samples, but their GP-MSE on unseen targets is 4–10$\times$ worse than ours across all four families. The decoded kernel does not memorize the context but it inherits the structural regularization of the PFN's pretraining prior, which is exactly the property the mechanistic analysis identified (see Figure 23). Moreover, the decoded kernel supports downstream Bayesian tasks that the PFN itself cannot perform without continuous GPU support. We obtained competitive Bayesian optimization performance by decoded kernel on four component spectral mixture with 0.006 ± 0.029 are average regret while oracle GP got 0.004 ± 0.028. Further, decoded kernel only uses CPU (Appendix G, Tables 15). Figure 21 shows true and decoded kernel matrices pictorially.

Multi-realization setting. With M independent realizations from the same prior, Theorem 2 guarantees full identifiability of $S(\omega)$, and the decoder realizes this in practice: the Wasserstein distance between decoded and ground-truth densities decreases monotonically with M across bandwidths (Fig. 18), with 1–4 component mixtures recovered faithfully (Fig. 19). On in-distribution spectral mixtures, decoded kernels match oracle GP MSE to within a factor of three (e.g. 1.14×10^{-4} vs. 1.45×10^{-4} on SM-$Q=2$), and degrade gracefully on out-of-distribution families (4.8×10^{-3} on RBF, vs. catastrophic failure for the sparse-spectral assumption). The same pattern holds under additive kernels in 5D and 10D, where the decoder stays within $\sim 3\times$ of the oracle on every spectral mixture family (Appendix F, Tables 12, 13). See Figure 20 for pictorial depiction of decoded kernel matrix.

6 Limitations and Implications

Limitations. Beyond the real-world time series validation (Section 4.3), tabular data patching (Section C.3) and 5D additive-GP experiments (Sec. 3.3, Appendix F), the probing experiments are restricted to stationary kernels representable as spectral mixtures and sinusoids, and the causal subspace analysis has not yet been extended to settings where noise, distributional shift, and active dimensionality interact simultaneously. The decoder is a diagnostic read-out, not a replacement for the PFN [Müller et al., 2022a]: Table 3 shows a consistent MSE gap, indicating that some predictive information escapes the spectral-mixture parametrization. Our causal account identifies *where* spectral information is stored and *that* it is used, but not *how* attention constructs it. We hypothesize that the MLP sub-layers apply a learned rectifying nonlinearity analogous to a periodogram's square-and-average step (Section 3.2), but verifying this via path patching remains future work.

Implications for PFN design. If spectral structure is constructed in a single cross-attention step (Sec. 4.1) and concentrated in a low-dimensional causal subspace (Sec. 4.2), most of a PFN's depth and width buys posterior formatting, not Bayesian capacity. The grid in Appendix K is consistent with this reading. Depth saturation tracks the $L_1 \to L_2$ emergence: at $d=128$ the $L=2 \to L=6$ gain is only $2.2\times$, at $d=48$ it shrinks to $1.2\times$ and ceases to be monotone (Fig. 24a) — the regime where a narrow residual cannot absorb additional formatting capacity. Width plateaus by $d \approx 48$–64, past which extra ambient dimensions do not enlarge the causal subspace of Sec. 4.2. The Pareto frontier (Fig. 24b) makes the consequence concrete: $d=64$, $L=2$, MQA reaches test MSE 7.7×10^{-5}, within $2.5\times$ of our largest configuration ($d=128$, $L=6$, Standard, 1.25M params) at $12\times$ fewer parameters and $3.2\times$ lower latency — the readout compresses, the construction does not. We read this as a sanity check that the mechanistic claims have design content, suggesting distillation and adapter-style tuning should target formatting layers rather than the first cross-attention step.

7 Conclusion

This paper provided mechanistic evidence that PFNs encode spectral structure in a linearly decodable, low-dimensional, and causally active subspace of the mean-pooled latent attention score $\bar{H}$. This structure emerges after a single attention step and generalizes across three architectures, including a frozen TabPFN probed with out-of-distribution inputs, indicating it is an emergent property of PFN-style amortization rather than because of any particular training prior, on regression tasks with continuous covariates. The Filter Bank Decoder further demonstrates that this latent structure is rich enough to reconstruct explicit stationary kernels via Bochner's theorem, yielding GP regression competitive with iterative baselines at lower latency. Together, these results show that PFN priors are not merely implicit: they are explicitly recoverable as portable Bayesian objects that support downstream tasks—including CPU-only Bayesian optimization, without re-invoking the network.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *OpenAI Blog*, 2023.

Matthias Bitzer, Mona Meister, and Christoph Zimmer. Amortized inference for Gaussian process hyperparameters of structured kernels. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 184–194. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/bitzer23a.html.

Salomon Bochner. *Lectures on Fourier Integrals*. Princeton University Press, Princeton, NJ, 1959.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=89ia77nZ8u.

David Duvenaud, James Robert Lloyd, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Structure discovery in nonparametric regression through compositional kernel search, 2013. URL https://arxiv.org/abs/1302.4922.

Batu El, Deepro Choudhury, Pietro Lio, and Chaitanya K. Joshi. Understanding information flow in graph transformers via attention graphs. In *ICLR 2025 Workshop: XAI4Science: From Understanding Model Behavior to Discovering New Scientific Knowledge*, 2025. URL https://openreview.net/forum?id=WpHMhLGOmj.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. Tunetables: context optimization for scalable prior-data fitted networks. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 160–187. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/geiger24a.html.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the state of the art in tabular foundation models, 2025. URL https://arxiv.org/abs/2511.08667.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2733–2743, 2019.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Siyang Liu and Han-Jia Ye. TabPFN unleashed: A scalable and effective solution to tabular classification problems. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5DD3RCcVcT.

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models, 2014. URL https://arxiv.org/abs/1402.4304.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Samuel Müller, Sebastian Pineda Arango, Matthias Feurer, Josif Grabocka, and Frank Hutter. Bayesian optimization with a neural network meta-learned on synthetic data only. In *Sixth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=9xCudkMSkC.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=KSugKcbNf9.

Samuel Müller et al. Position: The future of bayesian prediction is prior-fitted. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

Kaustubh Sharma, Simardeep Singh, and Parikshit Pareek. Decoupled-value attention for prior-data fitted networks: Gp inference for physical equations, 2025. URL https://arxiv.org/abs/2509.20950.

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, 2020. URL https://arxiv.org/abs/2006.10739.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*, 2023.

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1067–1075, 2013.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning, 2015. URL https://arxiv.org/abs/1511.02222.

Supplementary Material: Mechanistic Evidence for Spectral Structures in Prior-Data Fitted Networks

A Interpretability Experiments: Experimental Protocols

A.1 Data Generation

For all experiments, we generate sinusoidal signals on $t \in [-1, 1]$ with 200 points. Frequencies are sampled uniformly from $[0.5, 5.0]$ Hz with random phases $\phi \sim \mathcal{U}[0, 2\pi]$. For weighted signals,

$$y = a \cdot \sin(2\pi f_1 t + \phi_1) + (1 - a) \cdot \sin(2\pi f_2 t + \phi_2).$$

A.2 PFN Preprocessing

Following the PFN training protocol, we normalize inputs as:

$$Y_{\text{norm}} = \frac{Y - \mu}{\sigma}, \quad \text{then apply sigmoid activation:} \quad Y_{\text{proc}} = \sigma\left(0.75 \cdot Y_{\text{norm}}\right).$$

A.3 Probe Architectures and Training

Linear probe. A single ridge regression ($\alpha = 1.0$) with input $\bar{H} \in \mathbb{R}^d$ and scalar target. No hidden layers, d parameters. Inputs are standardized (StandardScaler) before fitting. Any R^2 this probe achieves is a lower bound on what is linearly encoded in $\bar{H}$.

MLP probe. A three-layer feed-forward network with hidden widths $256 \to 128 \to 64$. Each layer uses LayerNorm, GELU activation, and dropout ($p = 0.1$). Optimized with AdamW (learning rate 10^{-3}, weight decay 10^{-4}) using a cosine-annealing schedule (max 500 epochs, batch size 64) and early stopping on validation R^2 with patience 50.

Data and splits. Each probing set consists of 2000 synthetic signals (1000 for VA-PFN), each with 200 observation points on $t \in [-1, 1]$. Signals are passed through the frozen PFN and $\bar{H} = \frac{1}{N}\sum_i H[i]$ is mean-pooled over positions. Train /validation / test splits are 65% / 15% / 20%.

B Additional Mechanistic Results

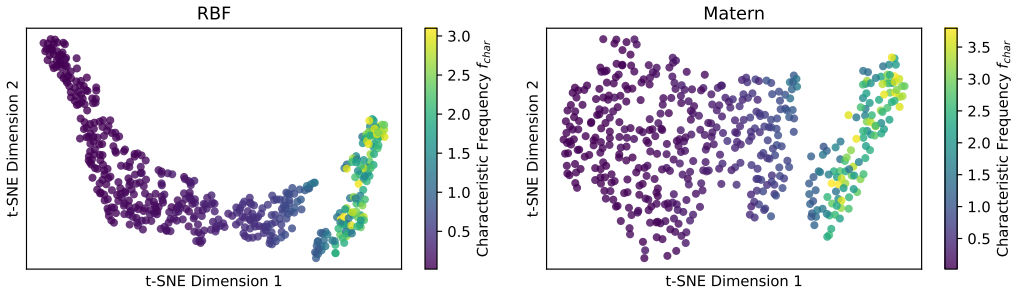
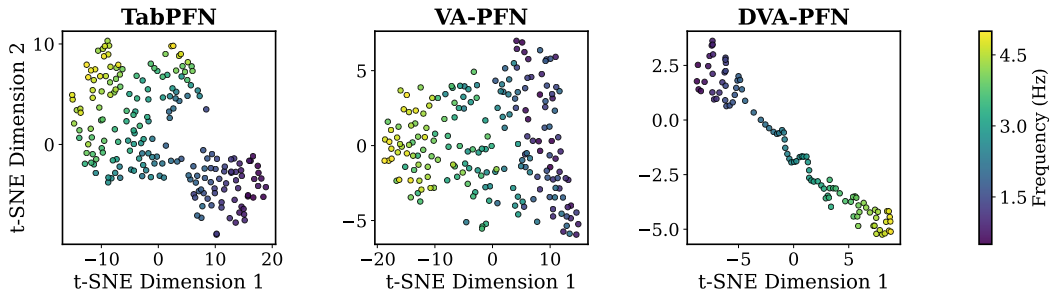
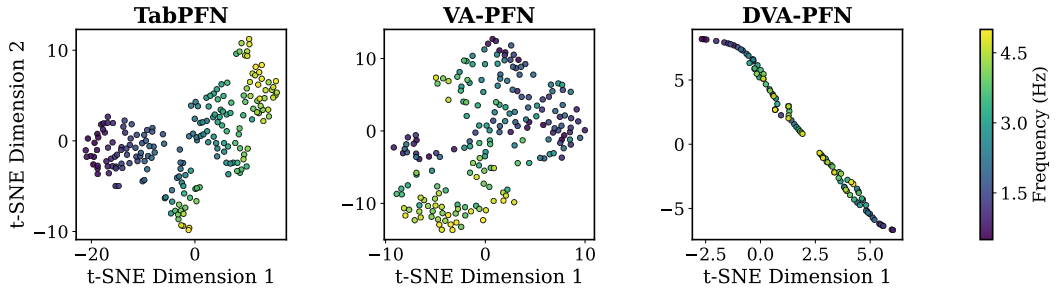


Figure 5: t-SNE projections of mean-pooled TabPFN embeddings $\bar{H}$ for 500 functions drawn from 5D GPs with RBF (left) and Matérn-3/2 (right) kernels, colored by characteristic frequency $f_{\text{char}} = 1/(2\pi\ell)$. Despite the absence of any sinusoidal structure in the generating process, embeddings organize smoothly by spectral scale, consistent with the quantitative metrics in Table 6.



(a) Mean-pooled attention weights $\bar{H}$ manifold.



(b) Mean-pooled value encoding $\bar{V}$ manifolds.

Figure 4: Manifold structures colored by frequency obtained via frequecy varying experiment described in Section 3. Note the tight spectral clusters in DVA-PFN compared to the smoother manifolds in VA and TabPFN due to diffusion of information inside attention mechanism.

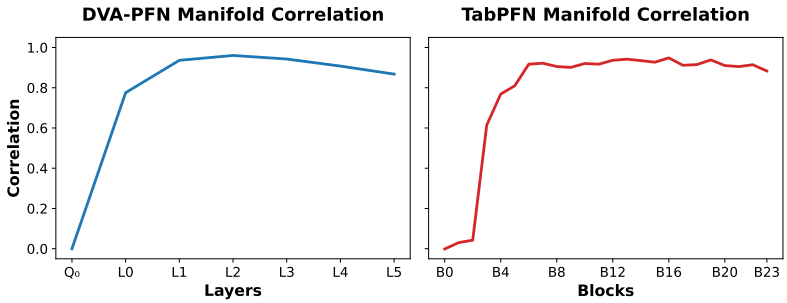


Figure 6: Layer-wise correlation ρ_Δ between embedding distances and generating-frequency distances. DVA-PFN (left) peaks at L2 ($\rho_\Delta = 0.95$) and TabPFN (right) plateaus near 0.93 by B12. Both architectures show a mild decline in the final layers, consistent with a shift toward posterior formatting.

Table 4: Pooling ablation on $\bar{H}$ (MLP probe, R^2) on DVA-PFN. Mean pooling degrades sharply as spectral complexity grows, while multi-query attention pooling recovers an increasing fraction of the lost signal. V-only probes yield $R^2 = 0$ at every difficulty and are omitted. Δ reports the gain of attention pooling over mean pooling on $H+V$.

Task	Mean(H)	Mean($H+V$)	Attn(H)	Attn($H+V$)	Δ
Easy (1 param)	0.999	1.000	1.000	1.000	+0.000
Medium (2 params)	0.982	0.983	0.991	0.991	+0.008
Hard (4 params)	0.560	0.564	0.610	0.602	+0.038
Very Hard (6 params)	0.333	0.342	0.404	0.399	+0.057

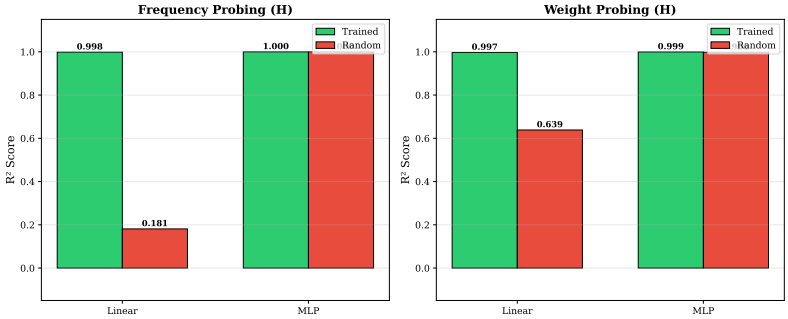


Figure 7: Control experiment: probing on trained vs. randomly initialized VA-PFN. Linear and MLP probes are trained to recover frequency (left) and mixing weight (right) from frozen $\bar{H}$. On the trained network both probes succeed ($R^2 \geq 0.99$). On the randomly initialized network, the MLP probe still recovers both targets—consistent with its capacity to learn the mapping itself [Hewitt and Liang, 2019]—but the linear probe fails ($R^2 = 0.18$ for frequency, 0.64 for weight), confirming that linear accessibility is a consequence of learned representational structure, not of trivial input geometry.

Table 5: Probing R^2 scores for spectral parameter extraction using **linear probes** on DVA-PFN. H consistently dominates V across all targets.

Target	H	V	$H + V$
Single Frequency	0.98	0.21	0.99
Dual Frequencies	0.96	0.00	0.96
Full Spectral (f_1, f_2, a_1, a_2)	0.50	0.00	0.50

Table 6: Parameter-free probing metrics for 5D GP functions on TabPFN.

	RBF		Matérn-3/2	
	ρ_Δ	$\lvert r\rvert_{\mathrm{PC0}}$	ρ_Δ	$\lvert r\rvert_{\mathrm{PC0}}$
TabPFN (5D)	0.815	0.900	0.861	0.928

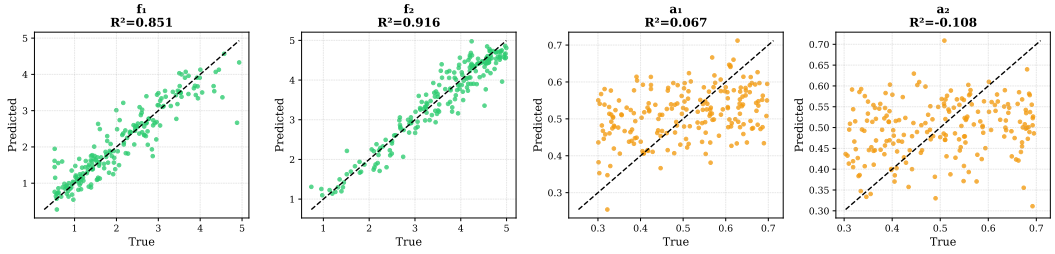


Figure 8: Probing performance on multi-component signals shows successful frequency extraction (f_1, f_2) but poor relative amplitude prediction (a_1, a_2) on TabPFN.

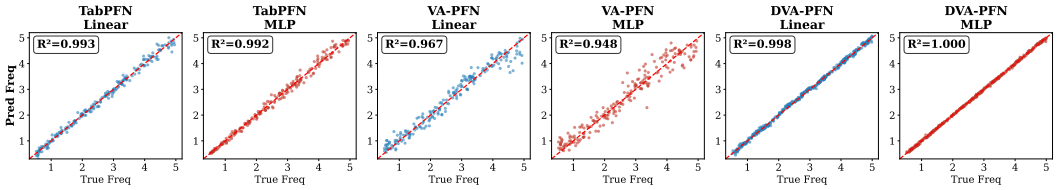


Figure 9: Frequency Probing: Linear and MLP Relational Scatter Plots across PFN architectures.

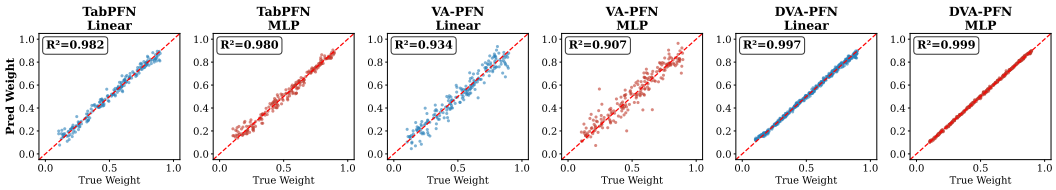


Figure 10: Weight Probing: Linear and MLP Relational Scatter Plots across PFN architectures.

C Mechanistic Analysis Results

C.1 Probing Comparison (Frequency and Mixing Weights)

We quantify the accessibility of spectral information by training linear and MLP probes on frozen latent representations. Figure 9 show the scatter plots for frequency prediction across the three models. Similarly, Figure 10 and show the results for mixing weight estimation in dual-component signals.

C.2 Causal Evidence (Dose-Response Patching)

Functions are drawn from a zero-mean GP with RBF kernel $k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2/\ell^2\right)$ on a fixed grid of N=200 equally-spaced points in $[-1, 1]$. Lengthscales are sampled log-uniformly: $\ell \sim \mathrm{LogUniform}(0.05, 2.0)$. All realisations use fixed random seeds, making the dataset fully deterministic and reproducible. Figure 11 shows representative pairs at the extremes of the ℓ range used in the activation patching experiment.

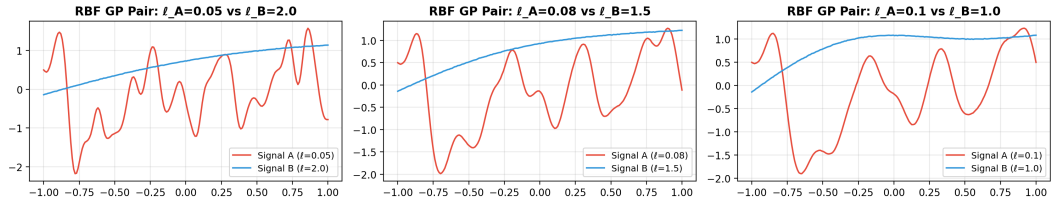


Figure 11: Representative RBF-GP signal pairs used for activation patching. Small ℓ (red) produces high-frequency wiggly functions; large ℓ (blue) produces smooth slowly-varying functions. The large visual separation ensures a strong baseline $\mathrm{MSE}(\hat{y}_A, \hat{y}_B)$, making the causal effect measurement well-conditioned.

Table 7 reports layer-wise causal effect (CE) for DVA-PFN under H-, V-, and K-patching, averaged over three signal pairs. CE is defined as

$$\mathrm{CE} = 1 - \frac{\mathrm{MSE}(\hat{y}_{\mathrm{patched}},\, \hat{y}_B)}{\mathrm{MSE}(\hat{y}_A,\, \hat{y}_B)}, \tag{2}$$

where $\hat{y}_A, \hat{y}_B$ are the unpatched model predictions for contexts y_A and y_B respectively (using model predictions rather than raw GP realisations removes sample-noise from the denominator).

Table 7: Layer-wise CE for DVA-PFN on RBF-GP pairs ($\ell_A = 0.05$, $\ell_B = 2.0$). H-patch reaches CE$\approx$1 by layer 2; K-patch gives exactly 0.

Patch type	L1	L2	L3	L4	L5	L6
H (causal)	0.00	**1.00**	1.00	1.00	1.00	1.00
V (+control)	1.00	1.00	1.00	1.00	1.00	1.00
K ($-$control)	0.00	0.00	0.00	0.00	0.00	0.00

Prior to the dose-response experiment, we verify that PCA identifies a meaningful spectral subspace.

- **DVA-PFN** (d=128): top-5 PC correlations with ℓ are $[0.30, 0.28, 0.26, 0.19, 0.18]$. Information is distributed across many PCs—consistent with DVA-PFN's narrower training distribution—yet concentrated enough that the top-64 PCs account for all causal transfer ($\mathrm{CE}_{\mathrm{spec},\, k=64} = 0.93$).

- **TabPFN** (d=192): PC_0 alone achieves $|r|$=0.882 and explains 73.6% of embedding variance. The remaining PCs fall off rapidly ($|r| \leq 0.21$), indicating a strongly dominant single axis of structural variation.

Table 8: Hyperparameters for non-sinusoidal patching experiments.

Parameter	Value
Input grid size N	200
Lengthscale range $[\ell_{\min}, \ell_{\max}]$	$[0.05,\ 2.0]$
Probe set size	300 signals (log-uniform ℓ)
Patch pairs	30 (min gap $\lvert\ell_A - \ell_B\rvert \geq 0.8$)
Subspace dims k	$\{1, 2, 4, 8, 16, 32, 64, 128\}$
DVA-PFN intervention layer	Cross-attention block 2 (of 6)
TabPFN intervention layer	Transformer block 24 (final)
Random seed	Fixed (all experiments deterministic)

Table 9: Layer-wise Causal Effect (CE) for activation patching. Mean $\pm$ s.d. over $n = 50$ pairs; p-values from paired t-tests (H vs. K).

Layer	**H-patch CE**	**V-patch CE**	**K-patch CE**	p **(H vs. K)**
L1	0.000 ± 0.000	0.999 ± 0.002	0.000 ± 0.000	n.a.
L2	0.999 ± 0.002	0.999 ± 0.002	0.000 ± 0.000	$< 10^{-133}$
L3–L6	0.999 ± 0.002	0.999 ± 0.002	0.000 ± 0.000	$< 10^{-133}$

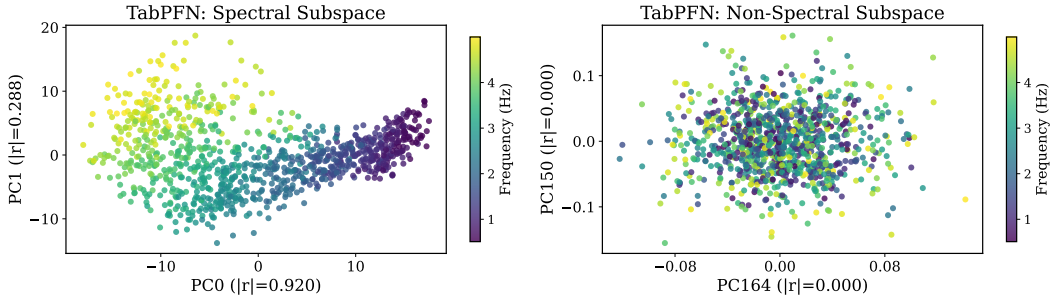


Figure 12: PCA of TabPFN embeddings

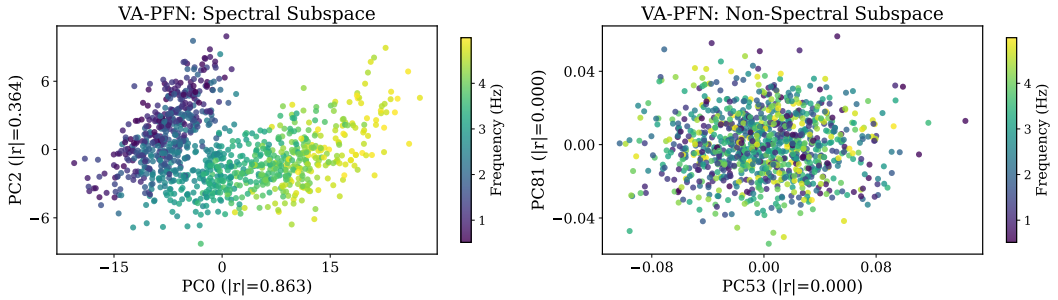


Figure 13: PCA of VA-PFN embeddings

C.3 Patching with Tabular Data

The patching experiments of Sections 4.1–4.2 use sinusoidal or GP-drawn signals that lie outside TabPFN's pretraining distribution. To verify that $\bar{H}$ causally encodes structural information on inputs

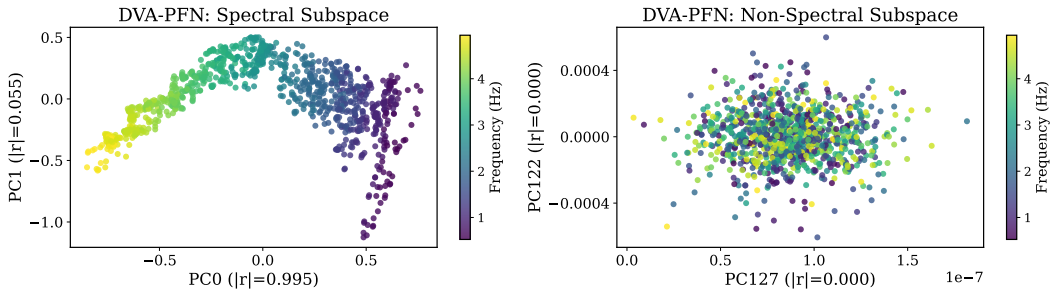


Figure 14: PCA of DVA-PFN embeddings

Table 10: Targeted subspace patching on DVA-PFN: causal effect as a function of subspace dimensionality k (out of $d = 128$). Selectivity = Spectral CE / Random CE.

k	% of dims	Spectral CE	Non-spectral CE	Random CE	Selectivity
1	0.78%	0.026	−0.000	0.007	—
2	1.56%	**0.592**	0.008	0.018	**33.9**×
4	3.13%	**0.676**	0.017	0.032	21.3×
8	6.25%	0.676	0.023	0.066	10.3×
16	12.5%	0.689	0.115	0.149	4.6×
32	25.0%	**0.977**	0.135	0.326	3.0×
128	100%	0.999	0.999	0.999	1.0×

TabPFN was designed for, we repeat the all-block patching protocol on **8-dimensional tabular data** sampled from $\mathcal{U}[0,1]^8$, consistent with TabPFN's synthetic pretraining regime. We fix a shared feature matrix $X \in \mathbb{R}^{100\times 8}$ and construct two targets from disjoint feature subsets: $y_A = f(x_0, x_1)$ and $y_B = f(x_5, x_6)$, where $f(x_i, x_j) = x_i \sin(4x_j) + x_i^2 + \varepsilon$ is a shared nonlinear function. As a negative control, y_B is replaced with a random permutation y_{rand} that destroys feature-relevance structure while preserving marginal statistics. Figure 17 reports causal effect averaged over 30 pairs with all 24 blocks patched simultaneously. $\bar{H}$-patching with a structurally different target achieves CE $= 0.985 \pm 0.014$, confirming near-complete transfer of feature-relevance identity through $\bar{H}$. The negative control yields CE $= 0.415 \pm 0.115$ which is substantially lower, though nonzero. This residual is expected under all-block replacement: even activations from a random target form an internally consistent signal that coherently overrides the base computation, shifting predictions away from $\hat{y}_A$ without specifically targeting $\hat{y}_B$. The decisive contrast where structured patching drives CE $\to 1$ while unstructured patching plateaus below 0.5, confirms that $\bar{H}$ causally encodes *which features drive the target*, extending the spectral-identity results of Sections 4.1–4.2 to TabPFN's native multi-feature regime.

C.4 Real-world Tabular Data Patching Details

Data. We use two publicly available monthly time series: *Airline Passengers* (Box & Jenkins, 144 observations, 1949–1960) loaded via `statsmodels`, and *Milk Production per cow* (USDA, 168 observations, 1962–1975). Both exhibit strong seasonality (period ≈ 12 months) but differ in trend structure and amplitude dynamics. Each series is first-differenced to remove trend, then z-normalized. A 5-lag embedding converts each to a tabular regression problem: $X_t = [y_{t-1}, \ldots, y_{t-5}] \in \mathbb{R}^5$, $\hat{y} = y_t$, yielding 138 (Airline) and 162 (Milk) samples respectively.

Model. We use `TabPFNRegressor` v2.5 on GPU with default hyperparameters. The model contains 24 transformer blocks; all interventions target block 23 (the final block). Baseline regression quality: $R^2 = 0.81$ (Airline) and $R^2 = 0.94$ (Milk) with 100 context and 30 test points.

Activation Patching (Exp. K2). For each ordered pair (source, donor), we:

1. Fit TabPFN on the source context and predict on 30 test points $\to \hat{y}_A$.
2. Fit on the donor context and predict $\to \hat{y}_B$.

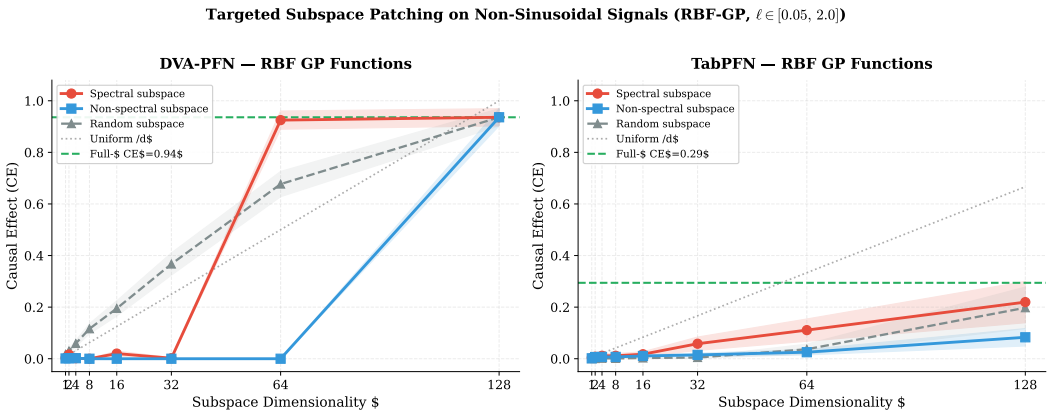


Figure 15: **Targeted subspace patching on RBF-GP functions.** Causal Effect (CE) as a function of patching dimensionality k for DVA-PFN (*left*) and TabPFN (*right*). **Red**: spectral subspace (PCs most correlated with ℓ). **Blue**: non-spectral subspace (bottom-k PCs). **Grey** triangles: random k-dimensional subspace. Dashed grey: uniform k/d baseline. Dashed green: full-H replacement ceiling. Shaded bands: 95% confidence intervals over 30 signal pairs. The spectral subspace dominates at every k for both architectures, demonstrating that causal structural information is compactly organised beyond the sinusoidal training distribution.

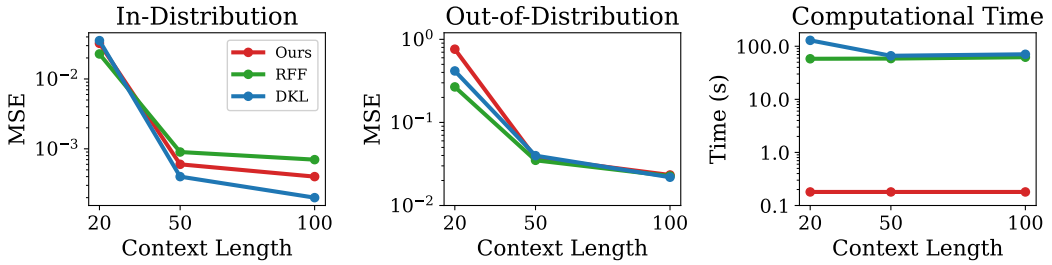


Figure 16: GP-MSE versus context length on (left) in-distribution sinusoids and (right) out-of-distribution triangle waves with random frequencies $\sim \mathcal{U}(1.0, 3.0)$. Decoded kernels match DKL/RFF on both, without per-task optimization.

3. Cache the full output tensor $H_B \in \mathbb{R}^{1\times194\times14\times192}$ at block 23 during the donor forward pass.
4. Re-fit on the source context, register a hook that replaces block 23's output with H_B, and predict $\to \hat{y}_{\text{patch}}$.
5. $\text{CE} = 1 - \text{MSE}(\hat{y}_{\text{patch}}, \hat{y}_B) \,/\, \text{MSE}(\hat{y}_A, \hat{y}_B)$.

The tensor dimensions correspond to batch (1), sequence positions (194 = context + test + padding), feature groups (14, internal to TabPFN), and model width (d=192). Shapes match across datasets because context size, test size, and input dimensionality are identical.

Negative control: replacing H with a random Gaussian tensor of the same shape. Results:

Direction	$\text{CE}_{\text{struct}}$	CE_{rand}
Airline ← Milk	**0.979**	0.013
Milk ← Airline	**0.891**	0.000

Subspace Patching (Exp. K3). To identify the "spectral" subspace, we bootstrap 100 context subsets per series, extract the mean-pooled $\bar{H} \in \mathbb{R}^{192}$ at block 23 for each, and fit PCA on the pooled 200-vector matrix. The leading PC achieves $|r|$=0.975 with the binary series label and explains 65.6% of variance; subsequent PCs drop below $|r|$=0.16.

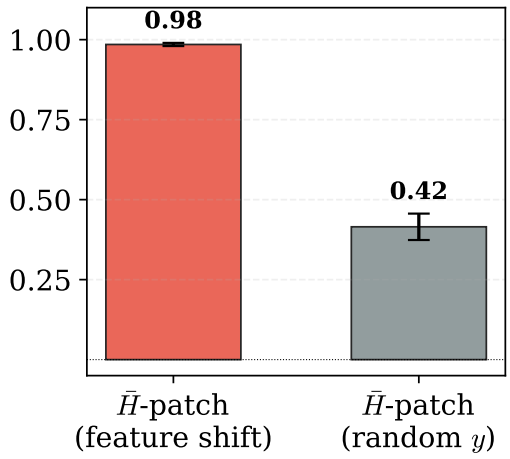


Figure 17: TabPFN patching on in-distribution 8D synthetic tabular data, comparing structured feature-relevance transfer against a random target control.

Dose-response patching operates on the *full tensor*: for a rank-k projection $\Pi = PP^\top \in \mathbb{R}^{192\times192}$ (with P containing the top-k or bottom-k PCs), the hook computes $H_{\text{patch}} = H_A - H_A\Pi + H_B\Pi$ applied element-wise along the last dimension of the $[1, 194, 14, 192]$ tensor.

k	Spectral	Non-spectral	Random
4	0.071	0.039	0.038
16	0.169	0.114	0.114
32	0.392	0.254	0.372
64	**0.798**	0.433	0.707
128	**0.964**	0.844	0.904
Full (192)		0.979	

The spectral subspace consistently outperforms the non-spectral subspace, achieving $\approx2\times$ the CE at k=64 and recovering 98.5% of the full-H ceiling at k=128. The random subspace tracks between the two, confirming that the PCA-identified directions carry disproportionate causal weight rather than the effect being purely dimensional.

D Statistical Identifiability of Spectral Density

We characterize what is fundamentally retrievable about the spectral density $S(\omega)$ of a stationary GP from observed function data, in two regimes: single-realization and multi-realization.

D.1 Single-Realization Limit

Theorem 1 (Single-function non-identifiability of spectral weights)**.** *Let $f \sim \mathcal{GP}(0, k)$ with continuous stationary kernel and spectral density $S(\omega) = \sum_{q=1}^{Q} w_q \mathcal{N}(\omega \mid \mu_q, \sigma_q^2)$, $w_q > 0$. Let $\{f(x_i)\}_{i=1}^{N}$ be a single realization on a fixed grid, normalized to unit empirical variance. Then $\{w_q\}_{q=1}^{Q}$ is not identifiable from this single realization, except up to a common multiplicative constant, even in the limit $N \to \infty$.*

Proof. Let $\mathbf{f} = (f(x_1), \ldots, f(x_N))^\top$ denote the vector of observations. Since $f \sim \mathcal{GP}(0, k)$ is stationary and Gaussian,

$$\mathbf{f} \sim \mathcal{N}(0, K), \qquad K_{ij} = k(x_i - x_j).$$

By Bochner's theorem, the kernel admits the spectral representation

$$k(\tau) = \int_{\mathbb{R}} e^{2\pi i \omega \tau} S(\omega)\, d\omega, \qquad S(\omega) = \sum_{q=1}^{Q} w_q \mathcal{N}(\omega \mid \mu_q, \sigma_q^2).$$

Consider any constant $c > 0$ and define a rescaled spectral density

$$\tilde{S}(\omega) = c\, S(\omega),$$

with corresponding kernel

$$\tilde{k}(\tau) = c\, k(\tau).$$

Let $\tilde{f} \sim \mathcal{GP}(0, \tilde{k})$. Then $\tilde{f}$ and f are related in distribution by

$$\tilde{f}(x) \overset{d}{=} \sqrt{c}\, f(x).$$

Let $\tilde{\mathbf{f}} = (\tilde{f}(x_1), \ldots, \tilde{f}(x_N))^\top$. Under empirical variance normalization which normalization mirrors the preprocessing used in PFNs, making the model invariant to global rescaling of function values.

$$\hat{\mathbf{f}} = \frac{\mathbf{f} - \bar{f}\mathbf{1}}{\|\mathbf{f} - \bar{f}\mathbf{1}\|_2}, \qquad \hat{\tilde{\mathbf{f}}} = \frac{\tilde{\mathbf{f}} - \bar{\tilde{f}}\mathbf{1}}{\|\tilde{\mathbf{f}} - \bar{\tilde{f}}\mathbf{1}\|_2}.$$

Since $\tilde{\mathbf{f}} = \sqrt{c}\,\mathbf{f}$, it follows immediately that

$$\hat{\tilde{\mathbf{f}}} = \hat{\mathbf{f}}.$$

Therefore, the normalized single-sample distribution induced by the kernel $k(\tau)$ is identical to that induced by $\tilde{k}(\tau) = ck(\tau)$. Because scaling the spectral density corresponds exactly to scaling all weights $\{w_q\}$ by the same constant c, no estimator operating on a single normalized realization can distinguish between $\{w_q\}$ and $\{cw_q\}$.Consequently, the spectral weights are not identifiable from a single realization, except up to a common multiplicative constant, even as $N \to \infty$. □

Remark. Frequency identifiability follows from classical spectral estimation theory: the discrete Fourier transform of a stationary process concentrates energy near the true frequencies, while global rescaling of the covariance affects only the magnitude, not the location, of spectral peaks.

Proposition 1 (Unbiased kernel-scale estimator)**.** *Let* $\mathbf{f} \sim \mathcal{N}(0, \alpha K)$ *for known PSD matrix* K *and unknown* $\alpha > 0$*. Then* $\hat{\alpha} = \|\mathbf{f}\|_2^2 / \mathrm{tr}(K_{pred})$ *satisfies* $\mathbb{E}[\hat{\alpha}] = \alpha$*.*

Proof. Let $\mathbf{f} = (f(x_1), \ldots, f(x_N))^\top \in \mathbb{R}^N$ denote the vector of function values evaluated at the fixed input locations $\{x_i\}_{i=1}^N$. Consider,

$$\mathbf{f} \sim \mathcal{N}(0, \alpha K_{\text{pred}}),$$

where $K_{\text{pred}} \in \mathbb{R}^{N \times N}$ is a fixed positive semi-definite matrix predicited using a single realization decoder and $\alpha > 0$ is an unknown scalar.

We define the squared ℓ_2 norm of $\mathbf{f}$ as $\|\mathbf{f}\|_2^2 = \mathbf{f}^\top \mathbf{f}$. All expectations below are taken with respect to the randomness of $\mathbf{f}$ induced by the Gaussian process, i.e. $\mathbb{E}[\cdot] = \mathbb{E}_{\mathbf{f} \sim \mathcal{N}(0, \alpha K_{\text{pred}})}[\cdot]$.

Since $\mathbf{f}$ is zero-mean Gaussian,

$$\mathbb{E}[\mathbf{f}\mathbf{f}^\top] = \alpha K_{\text{pred}}.$$

Taking the trace on both sides yields

$$\mathbb{E}[\|\mathbf{f}\|_2^2] = \mathbb{E}[\mathrm{tr}(\mathbf{f}\mathbf{f}^\top)] = \mathrm{tr}(\mathbb{E}[\mathbf{f}\mathbf{f}^\top]) = \alpha\, \mathrm{tr}(K_{\text{pred}}).$$

Therefore, for the estimator

$$\hat{\alpha} \;=\; \frac{\|\mathbf{f}\|_2^2}{\mathrm{tr}(K_{\text{pred}})},$$

we obtain

$$\mathbb{E}[\hat{\alpha}] = \frac{\mathbb{E}[\|\mathbf{f}\|_2^2]}{\mathrm{tr}(K_{\text{pred}})} = \alpha,$$

which proves that $\hat{\alpha}$ is an unbiased estimator of the kernel scale. □

D.2 Multi-Realization Guarantee

Theorem 2 (Identifiability from multiple realizations)**.** *Let $\{f_m\}_{m=1}^M$ be i.i.d. realizations from a zero-mean stationary GP with spectral density $S(\omega) = \sum_{q=1}^Q w_q \mathcal{N}(\omega \mid \mu_q, \sigma_q^2)$, $w_q > 0$, observed on a common fixed grid. As $M \to \infty$, the empirical covariance converges to $k(\tau)$ in probability, and $\{w_q\}_{q=1}^Q$ becomes identifiable from second-order statistics.*

Proof. We proceed by showing that multiple independent realizations allow consistent estimation of the covariance function, which uniquely determines the spectral weights.

For a zero-mean stationary Gaussian process, the covariance function

$$k(\tau) = \mathbb{E}[f(x) \cdot f(x+\tau)]$$

fully characterizes the process. By Bochner's theorem Rasmussen and Williams [2006], the covariance function $k(\tau)$ is in one-to-one correspondence with the spectral density $S(\omega)$. Therefore, identifying $k(\tau)$ is equivalent to identifying $S(\omega) = \sum_{q=1}^Q w_q \mathcal{N}(\omega \mid \mu_q, \sigma_q^2)$ and its parameters $\{w_q, \mu_q, \sigma_q\}$. We thus show that the empirical covariance converges to $k(\tau)$ as the number of realizations M increases.

Fix any pair of input locations (x_i, x_j) and define the empirical covariance estimator across realizations:

$$\hat{k}_M(x_i, x_j) \triangleq \frac{1}{M} \sum_{m=1}^M \big[f_m(x_i) \cdot f_m(x_j)\big].$$

Since the realizations $\{f_m\}$ are independent and identically distributed, each term $f_m(x_i) \cdot f_m(x_j)$ is an independent sample of a random variable with expectation

$$\mathbb{E}[f_m(x_i) \cdot f_m(x_j)] = k(x_i - x_j),$$

where the expectation is taken with respect to the Gaussian process prior.

Moreover, because $f_m(x_i) \cdot f_m(x_j)$ has finite second moment under the Gaussian process prior Rasmussen and Williams [2006], the Law of Large Numbers implies

$$\hat{k}_M(x_i, x_j) \xrightarrow{P} k(x_i - x_j) \quad \text{as } M \to \infty.$$

Since the input grid $\{x_i\}_{i=1}^N$ is fixed and finite, this convergence holds jointly for all pairs (i, j). Consequently, the entire empirical covariance matrix converges in probability to the true covariance matrix:

$$\big[\hat{k}_M(x_i, x_j)\big]_{i,j=1}^N \xrightarrow{P} \big[k(x_i - x_j)\big]_{i,j=1}^N.$$

Finally, the limiting covariance function $k(\tau)$ uniquely determines the spectral density $S(\omega)$ via Bochner's theorem. In particular, for the spectral mixture form

$$S(\omega) = \sum_{q=1}^Q w_q \mathcal{N}(\omega \mid \mu_q, \sigma_q^2),$$

the parameters $\{w_q\}$ are uniquely determined by $k(\tau)$. Therefore, as the number of independent realizations M increases, the spectral weights $\{w_q\}$ become identifiable from the empirical second-order statistics of the observed functions. □

E Filter Bank Decoder: Architecture and Training

E.1 Pipeline Overview

The decoder takes a context set $\mathcal{D}_{\text{ctx}}$, processes it through the frozen PFN to obtain H, V, and outputs explicit spectral parameters from which a stationary kernel is reconstructed via Bochner's theorem. The PFN is never updated.

E.2 Multi-Query Attention Pooling

For an input sequence $H \in \mathbb{R}^{B\times N\times d}$ and learned queries $Q \in \mathbb{R}^{1\times n_q\times d}$,

$$\mathrm{MQA}(H) \;=\; \mathrm{LinearFlatten}\big(\mathrm{MultiheadAttn}(Q,H,H)\big) \in \mathbb{R}^{B\times d}.$$

Independent MQA modules are applied to H and V, giving $z_H = \mathrm{MQA}_H(H)$, $z_V = \mathrm{MQA}_V(V)$, fused as $z = \mathrm{MLP}([z_H \,\|\, z_V])$.

E.3 Spectral Parameter Heads

We discretize the frequency range $[\mu_{\min}, \mu_{\max}]$ into B bins of width $\Delta = (\mu_{\max} - \mu_{\min})/B$. Three heads predict, per bin: (i) activation probability p_b, (ii) offset and bandwidth (δ_b, σ_b) giving $\mu_b = \mu_{\min} + (b + \delta_b)\Delta$, and (iii) weight w_b (multi-realization only; the single-realization decoder uses uniform $w_b = 1$ following Theorem 1).

E.4 Kernel Reconstruction

$$K(\tau) \;=\; \sum_{b:\,p_b>\gamma} w_b \exp\!\big(-2\pi^2\sigma_b^2\tau^2\big)\cos(2\pi\mu_b\tau), \qquad K_{\text{pred}} \;=\; \hat{\alpha}\,K, \tag{3}$$

with classification threshold γ and analytical scale $\hat{\alpha} = \|\mathbf{f}\|_2^2/\mathrm{tr}(K)$.

E.5 Loss and Curriculum

The decoder is trained with a composite loss

$$\mathcal{L} \;=\; \mathcal{L}_{\text{BCE}}(p, y_{\text{bin}}) \;+\; \lambda \sum_{b\in\text{active}} \|\theta_b - \theta_b^*\|^2, \tag{4}$$

with $w_{\text{pos}} = 30$ in BCE to handle sparse positive bins, and a curriculum that ramps n_p (active components) from 1 to 4. Hyperparameters in Table 11.

Table 11: Decoder training hyperparameters.

Parameter	Multi-Realization	Single-Realization
n_{samples}	100,000	300,000
n_{points}	200	200
n_{bins}	50	50
d_{model}	128	128
d_{ff}	256	256
n_{queries}	4	4
dropout	0.1	0.1
BCE positive weight	30.0	30.0
λ_{reg}	5.0	5.0
learning rate	10^{-3}	10^{-3}
weight decay	10^{-4}	10^{-4}
GP samples per task (M)	16	1
Frequency range (Hz)	$[0.5, 3.0]$	$[0.5, 3.0]$
σ range	$[0.01, 0.05]$	$[0.01, 0.05]$
Epochs (Phases 1 / 2 / 3)	1000 / 1000 / 2000	200 / 200 / 400

E.6 Training Data Generation

Multi-Realization. Spectral parameters: $\mu_q \sim \mathcal{U}[\mu_{\min}, \mu_{\max}]$, $\sigma_q \sim \mathcal{U}[\sigma_{\min}, \sigma_{\max}]$, $w_q \sim \mathrm{Gamma}(2,1)$. Kernel $K = \sum_q w_q \exp(-2\pi^2\sigma_q^2\tau^2)\cos(2\pi\mu_q\tau)$. GP samples $y^{(m)} \sim \mathcal{N}(0, K)$, $m = 1, \ldots, M$.

Single-Realization. Random Fourier Feature (RFF) signals

$$y(x) \;=\; \sqrt{\tfrac{2}{n_{\text{rff}} \cdot n_p}} \sum_{q=1}^{n_p} \sum_{j=1}^{n_{\text{rff}}} \cos(2\pi\omega_{qj} x + \phi_{qj}), \tag{5}$$

with $\omega_{qj} \sim \mathcal{N}(\mu_q, \sigma_q^2)$, $\phi_{qj} \sim \mathcal{U}[0, 2\pi]$, $n_{\text{rff}} = 100$.

F Additional Decoder Results

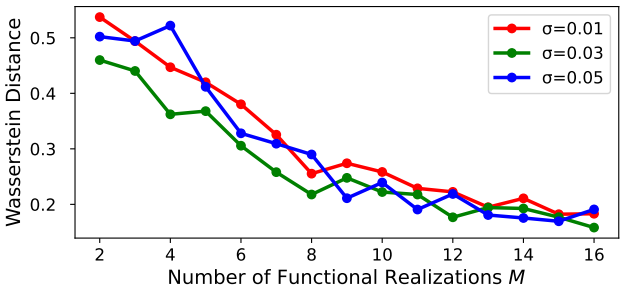


Figure 18: Wasserstein distance between true and decoded spectral densities as a function of the number of independent realizations M, for three bandwidths $\sigma \in \{0.01, 0.03, 0.05\}$. Monotone decrease is consistent with Theorem 2.

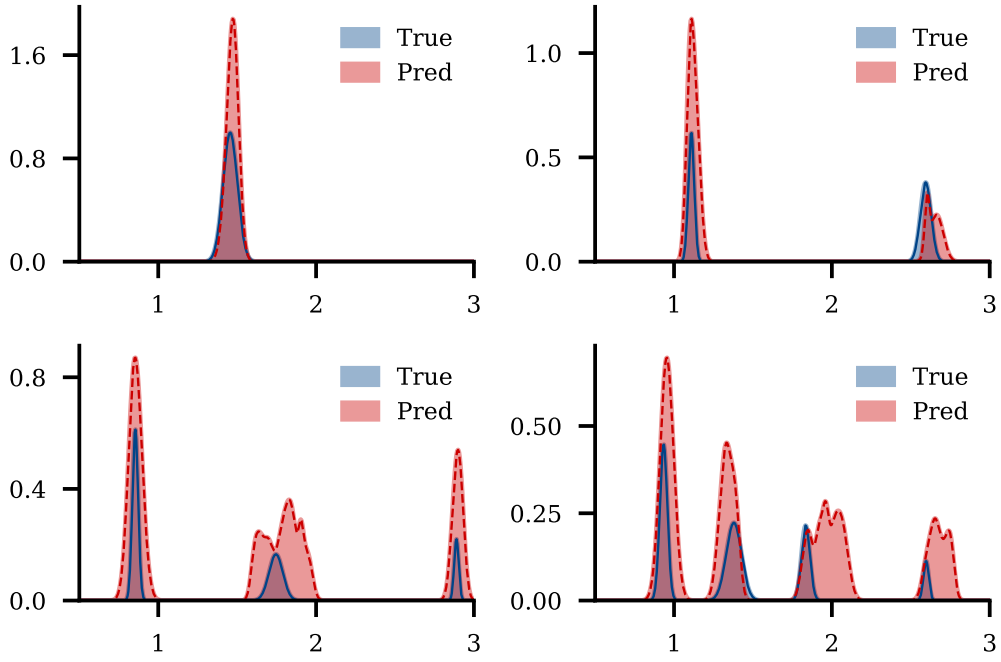


Figure 19: Decoded vs. ground-truth spectral densities for 1–4 component mixtures.

F.1 High-Dimensional Data Generation

Functions are drawn from additive GP priors $K(x, x') = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} K_d(x_d, x'_d)$, where $\mathcal{D}$ is a small active subset (2–4 in 5D; up to 6 in 10D under a curriculum) and each K_d is RBF, Periodic, or SM. Inputs are drawn independently per dimension and sorted along each coordinate, removing combinatorial spatial variation while preserving kernel structure.

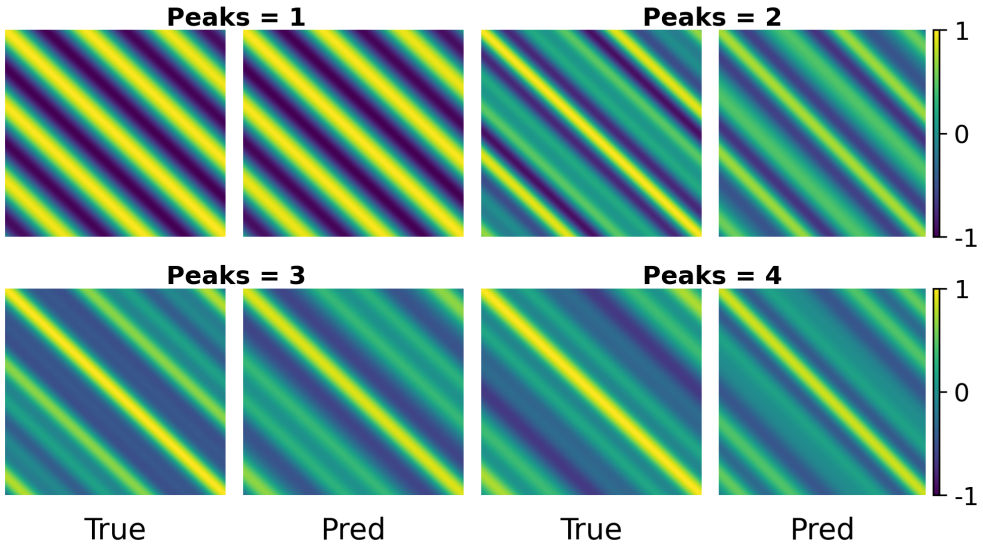


Figure 20: Kernel reconstruction under **multi-realization settings**. Left of each pair: ground-truth covariance. Right: decoded K_{pred}. Rows show 1–4 spectral peaks.

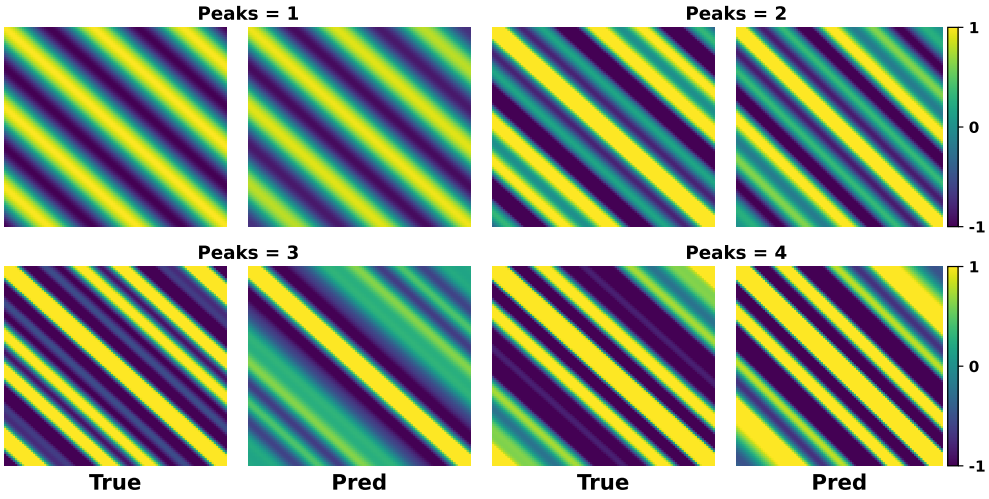


Figure 21: Kernel reconstruction from **single function observations**. Global amplitude is ambiguous (Theorem 1); dominant periodic structure and lengthscales are recovered.

Estimators.

$$\text{Periodogram:} \qquad P(\omega) = \tfrac{1}{N}\left|\textstyle\sum_{n=1}^{N} y_n e^{-i\omega x_n}\right|^2, \tag{6}$$

$$\text{Lomb–Scargle:} \qquad P_{\text{LS}}(\omega) = \tfrac{1}{2\sigma^2}\left[\frac{(\sum_n y_n \cos\omega(x_n-\tau))^2}{\sum_n \cos^2\omega(x_n-\tau)} + \frac{(\sum_n y_n \sin\omega(x_n-\tau))^2}{\sum_n \sin^2\omega(x_n-\tau)}\right], \tag{7}$$

where τ is the standard frequency-dependent time delay.

Why classical methods fail at generalization. Classical estimators achieve very low Ker-MSE on the context set — they memorize the discrete-sample artifacts, noise, and phase alignments of the specific realization. Theorem 1 predicts this: the weights are ambiguous from a single realization, and least-squares fits seize on whatever assignment best matches the observed samples. Once tested on unseen target locations, this overfit collapses, producing GP-MSE that is 4–10$\times$ worse than the PFN decoder across all four families. The decoder sidesteps this by inheriting the structural prior the

Table 12: Decoder GP-MSE on a fixed support of 16 functions per kernel, evaluated on 20 unseen test functions from the same prior. RBF and Matérn families lie outside the sparse-spectral-mixture model class; the decoder shows graceful degradation rather than failure.

Kernel family	Oracle GP MSE	Decoder MSE
SM ($Q=1$)	1.13×10^{-4}	6.15×10^{-4}
SM ($Q=2$)	1.45×10^{-4}	1.14×10^{-4}
SM ($Q=4$)	1.17×10^{-4}	2.99×10^{-4}
Out-of-distribution		
RBF	1.34×10^{-4}	4.76×10^{-3}
Matérn-1/2	9.50×10^{-2}	1.78×10^{-1}
Matérn-3/2	1.14×10^{-1}	2.21×10^{-1}
Matérn-5/2	1.61×10^{-1}	2.94×10^{-1}

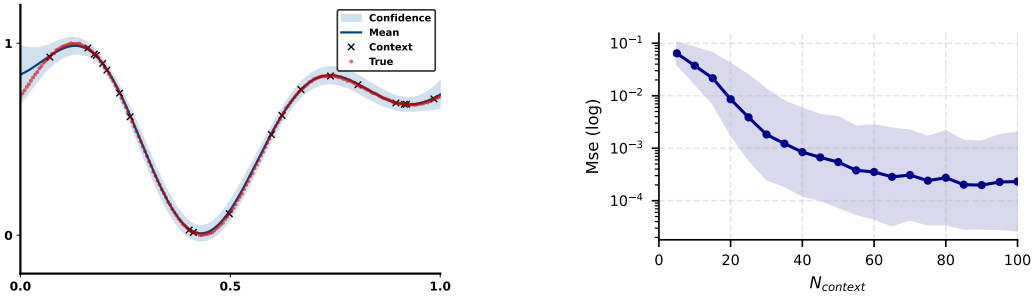


Figure 22: **Single-realization** decoder performance. (a) Qualitative GP regression from a decoded kernel with $N = 20$ context points. (b) GP-MSE versus context size N_{context}. The gap to the oracle remains roughly constant in N, consistent with the single-realization weight ambiguity (Theorem 1).

PFN learned during pretraining: peak locations and bandwidths come from a frozen representation that has already integrated over many realizations, so they generalize to new target points rather than tracking the context.

G Downstream Use: Bayesian Optimization with the Decoded Kernel

The decoded kernel parameterizes a standard GP that can be evaluated and updated without re-invoking the PFN. We illustrate this with a Bayesian optimization pipeline: PFN $\to$ decoder $\to$ SM kernel $\to$ GP posterior $\to$ UCB acquisition. The PFN runs once on the initial context to extract $k(\tau)$; subsequent BO iterations use only the decoded kernel.

Setup. 100 objective functions per kernel type (SM-$Q1$, SM-$Q2$, SM-$Q4$); $N_0 = 15$ initial observations; 50 BO iterations; UCB acquisition.

What this shows. The decoded kernel matches the oracle on simple mixtures and stays within one order of magnitude on $Q = 4$, while the entire BO loop runs on CPU at $\sim$5 ms per step. The PFN itself cannot be used this way: it produces predictive distributions at queried locations but does not expose a closed-form posterior, so it cannot supply the acquisition function or perform incremental posterior updates. The decoded kernel does both, validating the practical claim that the recovered Bayesian object is not just inspectable but reusable.

H High-Dimensional Data Generation

Functions are generated from additive GP priors $K(x, x') = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} K_d(x_d, x'_d)$, where $\mathcal{D}$ is the set of active dimensions and each K_d is a 1D kernel (RBF, periodic, or spectral mixture). For

Table 13: Multi-realization decoder GP-MSE in 5D and 10D additive-kernel settings, $M = 16$ functions per task. Predictions remain within a small constant factor of the oracle except on Periodic, which is hardest under additive structure.

Kernel (additive)	Oracle GP MSE	Decoder MSE
RBF (5D)	1.30×10^{-4}	2.90×10^{-4}
SM ($Q=1$, 5D)	1.10×10^{-4}	3.00×10^{-4}
SM ($Q=2$, 5D)	1.20×10^{-4}	2.40×10^{-4}
SM ($Q=4$, 5D)	1.30×10^{-4}	2.90×10^{-4}
Periodic (5D)	2.60×10^{-4}	9.50×10^{-3}
RBF (10D)	1.30×10^{-4}	3.25×10^{-4}
SM ($Q=1$, 10D)	1.15×10^{-4}	4.01×10^{-4}
SM ($Q=2$, 10D)	3.19×10^{-4}	5.38×10^{-4}
SM ($Q=4$, 10D)	2.06×10^{-4}	4.56×10^{-4}
Periodic (10D)	2.71×10^{-4}	2.32×10^{-3}

Table 14: Spectral recovery benchmark (50 context, 150 target points). Classical methods overfit the context set (low Ker-MSE) but fail at GP-MSE on unseen targets. The PFN-based decoder uses pretraining-induced regularization to generalize. We report Mean MSE over 100 samples. **Bold** marks best per column within each kernel family.

Kernel family	Method	Ker-MSE (↓)	Ker-CKA (↑)	GP-MSE (↓)
RBF	Periodogram	0.0060	**0.7316**	0.0859
	Lomb–Scargle	**0.0031**	0.7177	0.0788
	PFN Decoder (Ours)	0.0062	0.3648	0.0011
Periodic	Periodogram	0.0028	**0.8768**	0.0964
	Lomb–Scargle	**0.0023**	0.7191	0.0468
	PFN Decoder (Ours)	0.0031	0.5791	**0.0015**
Locally Periodic	Periodogram	0.0012	**0.7730**	0.0711
	Lomb–Scargle	**0.0008**	0.7118	0.0467
	PFN Decoder (Ours)	0.0012	0.6378	**0.0027**
Spectral Mixture	Periodogram	0.0008	**0.7870**	0.1174
	Lomb–Scargle	**0.0005**	0.7676	0.0989
	PFN Decoder (Ours)	0.0008	0.7007	**0.0017**

each function only 2–4 dimensions are active in 5D and up to 6 in 10D, ensuring well-conditioned covariance matrices. Inputs $x_d \in [0, 1]$ are sampled uniformly per dimension and sorted before kernel evaluation, removing combinatorial spatial variation while preserving the structure each K_d induces. Multiplicative kernel composition is avoided in high dimensions because it collapses covariance structure.

The decoder architecture is unchanged across dimensions; in 10D we use a four-phase curriculum on the number of active dimensions ($2\to3\to4\to\leq6$). PFN training in 5D and 10D uses the same architecture and hyperparameters as in 1D, with additive priors only.

I High-Dimensional Scaling Details

Functions are generated from additive Gaussian process priors of the form

$$K(x, x') = \frac{1}{|\mathcal{D}|} \sum_{d\in\mathcal{D}} K_d(x_d, x'_d),$$

where $\mathcal{D}$ denotes the set of active dimensions and each K_d is a one-dimensional kernel (RBF, periodic, or spectral mixture). Only a small subset of dimensions is active for any given function (between 2 and 4 in 5D), ensuring well-conditioned covariance matrices.

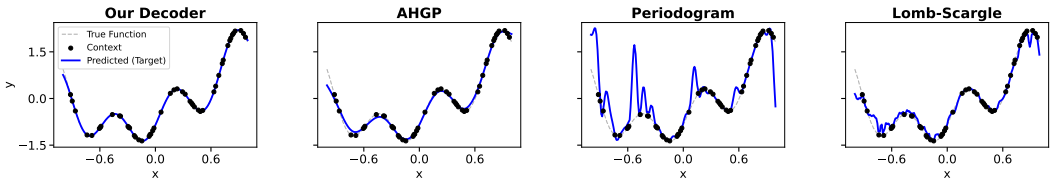


Figure 23: GP predictive posterior on a complex spectral mixture from a singlerealization. Periodogram and Lomb–Scargle interpolate the 50 context points exactly but oscillate wildly between observations; the decoded kernel recovers the underlying structure and tracks the held-out target.

Table 15: Resource comparison for iterative inference on CPU.

	PFN (per step)	Decoded Kernel + GP
Parameters	859,108	9
Memory	3.4 MB	72 B
Time / iteration (CPU)	24.8 ms	4.8 ms
GPU required	yes	no

J PFN Pre-Training Details

The DVA and VA PFN is trained on synthetic 1D functions drawn from a stationary GP with a random Spectral Mixture Kernel (SMK). The architecture is a Transformer-based Conditional Neural Process with 6 cross-attention blocks (4 heads, $d_{\text{model}} = 128$), trained using categorical cross-entropy over 100 discretized output bins. Our training pipeline is direct extention of the codes given by [Müller et al., 2022b] and [Sharma et al., 2025]

J.1 Baseline Implementation Details

Deep Kernel Learning (DKL). Feature extractor: 3-layer MLP ($1 \to 64 \to 64 \to 2$), Tanh hidden activations, linear output layer. GP: `ScaleKernel(RBFKernel(ard_num_dims=2))` on the 2D latent. Joint MLP and GP optimization via Adam (lr$= 0.01$, 500 iterations, exact MLL).

Random Fourier Features (RFF). Feature map: $\phi(x) = \sqrt{2/n}\cos(xW^\top/\ell + b)$, $n = 128$. $W \sim \mathcal{N}(0, I)$ and $b \sim \mathcal{U}[0, 2\pi]$ are fixed (non-optimized) buffers. Learnable: softplus-constrained ℓ, output scale, noise variance. GP: `ScaleKernel(LinearKernel())` on $\phi(x)$. Same optimizer, budget, and standardization as DKL.

GP Oracle. Exact GP with ground-truth kernel family, `GaussianLikelihood`, and `SpectralMixtureKernel` initialized via `initialize_from_data` for SM-Q tasks. Five restarts per task (Adam, lr$= 0.1$, 500 iterations); lowest MLL selected.

K Architecture Ablation Validating Mechanistic Predictions

Setup. We sweep the architecture along three axes using DVA-PFN: width $d \in \{16, 32, 48, 64, 96, 128\}$, depth $L \in \{2, 3, 6\}$, and attention variant (joint multi-head Standard vs. multi-query MQA), training each of the 36 combinations with 3 random seeds for a total of 108 runs. All runs share the training prior and pre-processing of Sec. K (hierarchical spectral mixture, sigmoid-normalized targets), the same optimizer (AdamW, $\text{lr} = 10^{-4}$, 5-epoch warm-up, cosine schedule), the same training budget (200 epochs of 100 steps each, batch size 32, randomized context length $n_{\text{ctx}} \in [100, 110]$), and the same frozen evaluation sets (500-function validation, 2000-function test) generated once with a fixed seed and loaded by every worker. Two architectural details are scaled with d to keep cross-cell comparisons fair. The FFN hidden width is set to $4d$ rather than fixed at 256, which would silently inflate small models (a fixed-FFN ablation is reported below). The head count is chosen so head dimension is in $[8, 16]$ regardless of d, rather than fixed at 4, which would collapse head dimension at $d = 16$. Latency is measured in ms per batch of 32 on a single GPU after 30 warm-up forward passes. The full grid summary is reported in Tab. 16.

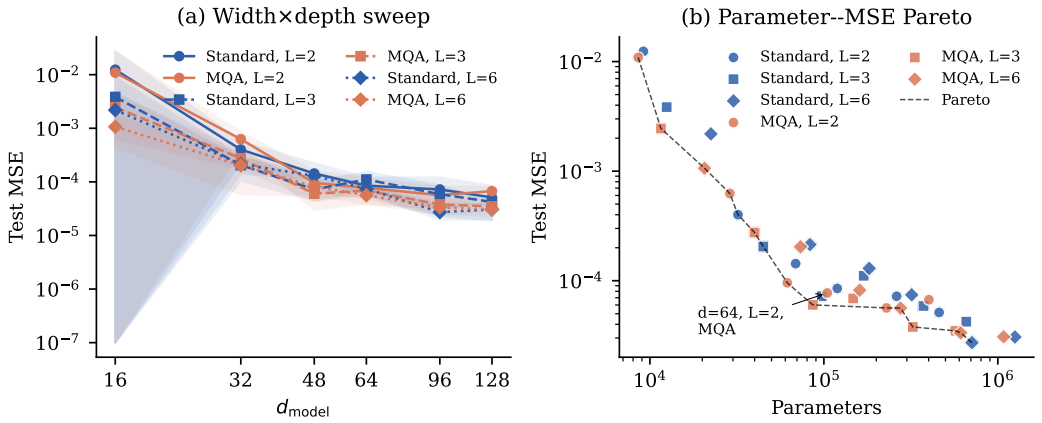


Figure 24: Architecture ablation supporting the mechanistic predictions of Secs. 3.2–4. Test MSE on the held-out 2000-function set across a $6\times3\times2$ grid of widths d, depths L, and attention variants (3 seeds each). **(a)** MSE vs. d_{model}, with shaded ±1 std bands across seeds. Width gains plateau by $d\approx48$–64, and the spread between depths shrinks as d shrinks — consistent with the $L_1\to L_2$ emergence of spectral coding (Sec. 4.1): once the first cross-attention step has constructed the spectrally-organized latent, additional layers contribute mainly posterior formatting, with diminishing returns. **(b)** Parameter–MSE Pareto frontier (dashed). The annotated configuration ($d=64$, $L=2$, MQA) achieves test MSE 7.7×10^{-5} at $\sim$105K parameters and 1.15 ms/batch — within $2.5\times$ of the largest configuration ($d=128$, $L=6$, Standard; 1.25M parameters, 3.66 ms) at $12\times$ fewer parameters and $3.2\times$ lower latency. 12 of 15 Pareto-optimal points use MQA. Full numerical table in Table 16.

Mechanistic predictions tested. Two findings of Secs. 3.2–4 generate concrete, falsifiable predictions about architecture. First, the abrupt $L_1\to L_2$ emergence reported in Sec. 4.1 where the causal effect of patching $\bar{H}$ jumps from 0 to ≈1 in a single cross-attention step — predicts that depth beyond two cross-attention layers should yield diminishing returns for spectrally-driven prediction, with the marginal benefit of additional layers attributable to posterior *formatting* rather than spectral *construction*. Second, the subspace concentration of Fig. 1, where roughly 10% of principal components account for the full causal effect, predicts that the residual-stream dimensionality required to carry the spectral computation is small in absolute terms, although learning to organize that one-dimensional signal in the right direction may demand a larger working budget than the signal itself occupies.

Depth saturates rapidly, consistent with $L_1\to L_2$ emergence. Holding width fixed and varying L (Table 16), depth gains shrink as d shrinks. At $d=128$ (MQA), increasing L from 2 to 6 roughly halves the test MSE ($6.7\times10^{-5}\to3.1\times10^{-5}$, a factor of 2.2). At $d=64$ the same $3\times$ depth increase yields only $1.4\times$; at $d=48$ it yields $1.2\times$ and the relationship is not monotone ($L=3$ outperforms both $L=2$ and $L=6$). We read this as the experimental dual of the manifold-correlation curves of Fig. 6: the first cross-attention layer produces the spectrally-organized $\bar{H}$, and additional layers chiefly perform downstream formatting — useful but with diminishing returns, especially when the residual stream is narrow enough that the formatting capacity is already saturated. The fact that depth helps *more* at $d=128$ than at $d=48$ also supports the formatting interpretation: a wider stream gives later layers more usable capacity to refine.

Headline configuration and limitations. The most efficient configuration that still approaches the full-capacity performance is $d=64$, $L=2$, MQA: test MSE 7.7×10^{-5}, $\sim$105K parameters, 1.15 ms per batch of 32. Our largest configuration ($d=128$, $L=6$, Standard) achieves 3.1×10^{-5} at ~1.25M parameters and 3.66 ms, so the small model trades $2.5\times$ MSE for $12\times$ fewer parameters and $3.2\times$ lower latency. We note that the benchmark is one-dimensional synthetic data drawn from a fixed prior family; the model is a small DVA-PFN, not a production-scale TabPFN. The ratios reported here should not be ported uncritically to higher dimensions or to TabPFN-scale architectures. Therefore, we present these number in Table 16 and Figure 24 as evidence that the mechanistic story has design content, not as a prescription.

Table 16: Full architecture grid (Appendix K). Six widths $\times$ three depths $\times$ two attention variants, three seeds each. Mean and standard deviation of test MSE across seeds; latency reported in ms per batch of 32 on a single GPU.

d	L	attn	params	MSE (mean)	MSE (std)	lat (ms)
16	2	MQA	8.6K	1.09×10^{-2}	1.58×10^{-2}	0.89
32	2	MQA	28.8K	6.26×10^{-4}	2.17×10^{-4}	0.80
48	2	MQA	61.6K	9.57×10^{-5}	6.54×10^{-5}	0.82
64	2	MQA	104.6K	7.74×10^{-5}	3.83×10^{-5}	1.15
96	2	MQA	229.1K	5.65×10^{-5}	2.47×10^{-5}	1.06
128	2	MQA	401.6K	6.71×10^{-5}	1.37×10^{-5}	1.20
16	3	MQA	11.6K	2.45×10^{-3}	1.80×10^{-3}	0.97
32	3	MQA	39.9K	2.75×10^{-4}	9.81×10^{-5}	1.20
48	3	MQA	86.4K	6.02×10^{-5}	1.58×10^{-5}	1.40
64	3	MQA	147.4K	6.89×10^{-5}	1.09×10^{-5}	1.57
96	3	MQA	324.7K	3.78×10^{-5}	1.37×10^{-5}	1.58
128	3	MQA	571.1K	3.49×10^{-5}	5.73×10^{-6}	1.95
16	6	MQA	20.6K	1.06×10^{-3}	6.17×10^{-4}	1.90
32	6	MQA	73.3K	2.04×10^{-4}	1.44×10^{-4}	2.44
48	6	MQA	160.7K	8.20×10^{-5}	2.64×10^{-5}	2.87
64	6	MQA	275.6K	5.63×10^{-5}	1.71×10^{-5}	3.46
96	6	MQA	611.5K	3.35×10^{-5}	7.72×10^{-6}	2.61
128	6	MQA	1079.5K	3.08×10^{-5}	7.75×10^{-6}	1.93
16	2	Standard	9.2K	1.24×10^{-2}	1.48×10^{-2}	0.75
32	2	Standard	32.1K	4.01×10^{-4}	1.66×10^{-4}	0.77
48	2	Standard	68.8K	1.43×10^{-4}	7.92×10^{-5}	0.97
64	2	Standard	119.3K	8.50×10^{-5}	3.52×10^{-5}	1.41
96	2	Standard	261.9K	7.21×10^{-5}	4.87×10^{-5}	1.22
128	2	Standard	459.7K	5.13×10^{-5}	3.24×10^{-5}	1.38
16	3	Standard	12.5K	3.84×10^{-3}	5.05×10^{-3}	1.12
32	3	Standard	44.8K	2.05×10^{-4}	9.91×10^{-5}	1.26
48	3	Standard	97.2K	7.22×10^{-5}	2.74×10^{-5}	1.33
64	3	Standard	169.4K	1.11×10^{-4}	4.26×10^{-5}	1.65
96	3	Standard	373.9K	5.88×10^{-5}	2.70×10^{-5}	1.74
128	3	Standard	658.3K	4.23×10^{-5}	2.28×10^{-6}	2.28
16	6	Standard	22.4K	2.19×10^{-3}	2.47×10^{-3}	2.23
32	6	Standard	83.1K	2.14×10^{-4}	7.05×10^{-5}	2.56
48	6	Standard	182.3K	1.30×10^{-4}	1.82×10^{-5}	3.12
64	6	Standard	319.8K	7.44×10^{-5}	1.50×10^{-5}	3.33
96	6	Standard	710.0K	2.72×10^{-5}	7.18×10^{-6}	3.66
128	6	Standard	1253.9K	3.06×10^{-5}	1.06×10^{-5}	3.66