ADAPTIVE MULTI-FREQUENCY REPRESENTATIONS FOR PHYSICS-INFORMED NEURAL NETWORKS WITH STRUCTURED VARIABLE SEPARATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural networks (PINNs) often struggle with multiscale, high-dimensional partial differential equations (PDEs) because of spectral bias and inefficient representations of interactions among variables. We propose adaptive multi-frequency representations for PINNs with structured variable separation (MFAR-SVS). Each neuron combines multiple sinusoidal branches with jointly learnable frequencies and weights, allowing its spectral representation to adapt to the target solution's frequency content rather than relying on fixed or manually selected frequencies. To exploit structured dependencies in high-dimensional solutions, we introduce a bipartite low-rank factorization that preserves unrestricted interactions within each variable group while imposing low-rank coupling only across groups. This provides an intermediate representation between fully coupled networks and dimension-wise tensor decompositions. We derive a closed-form neural tangent kernel (NTK) for the resulting architecture and characterize its infinite-width and infinite-rank limits. Theoretical analysis shows that adaptive frequency learning introduces a positive-semidefinite correction to the fixed-frequency kernel, implying non-decreasing ordered eigenvalues, including the minimum eigenvalue. Controlled experiments further show that adaptive frequencies progressively capture higher-frequency components and yield larger minimum NTK eigenvalues. Across four PDE benchmarks, learnable frequencies never worsen a decomposition and the structured separation is most accurate on the strongly coupled problems.

1 INTRODUCTION

Numerical solutions of partial differential equations (PDEs) are fundamental to scientific computing and engineering. Classical grid-based methods suffer from the curse of dimensionality, with computational cost growing rapidly with the number of variables (Strang & Fix, 2008; Strikwerda, 2004; Bellman, 1961; Bungartz & Griebel, 2004). Physics-informed neural networks (PINNs) (Raissi et al., 2019; Karniadakis et al., 2021) provide a mesh-free alternative by incorporating governing equations and physical constraints into neural network training Lu et al. (2021); Cuomo et al. (2022). However, their performance on high-dimensional and multi-scale PDEs critically depends on how the solution is represented.

One key challenge is the *frequency representation* of coordinate-based networks. Standard neural networks exhibit spectral bias, learning low-frequency components before high-frequency ones (Rahaman et al., 2019; Wang et al., 2021a;b). SIREN (Sitzmann et al., 2020) and Fourier features (Tancik et al., 2020) improve the representation of oscillatory functions, but their frequency content is typically fixed or manually specified, limiting adaptation to PDE solutions with unknown or multi-scale spectral components. A second challenge is the *structure of variable interactions*. Tensor-based PINNs, including separable, CP, TT, and Tucker formulations (Cho et al., 2023; Vemuri et al., 2024), reduce the cost of high-dimensional representations by exploiting low-rank structure, but impose different restrictions on variable coupling. Fully dimension-wise decompositions may separate physically correlated variables. In many spatiotemporal PDEs, spatial variables are more strongly coupled with each other than with time; for example, the spatial coordinates (x, y)

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

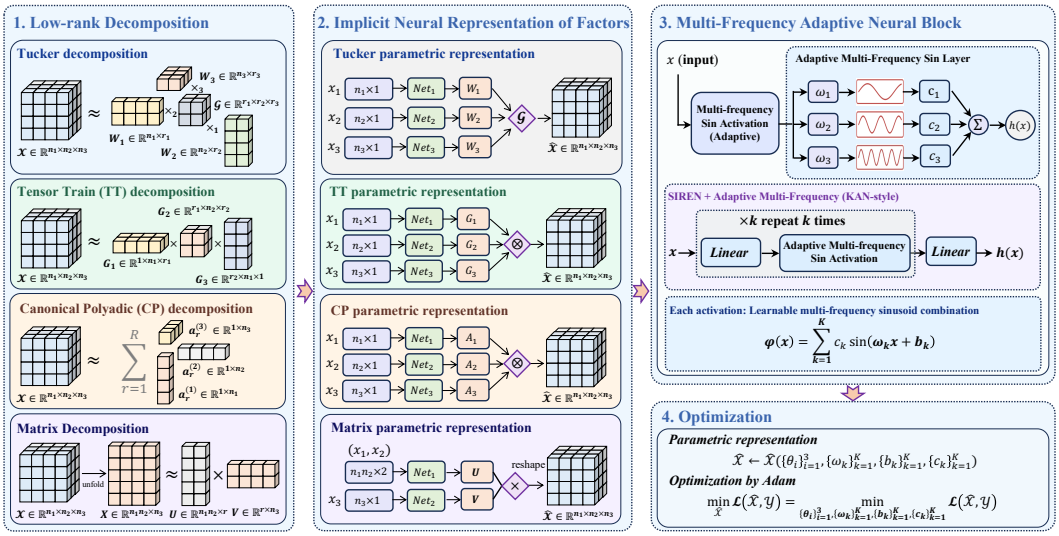


Figure 1: Overview of the proposed representation framework. **(a)** Low-rank decompositions, including matrix, CP, TT, and Tucker representations, expose structured dependencies in high-dimensional solutions. **(b)** These decompositions are parameterized by coordinate-based neural networks, with each factor represented as a function of its associated variables. **(c)** Adaptive multi-frequency representations replace fixed-frequency activations by combining multiple sinusoidal branches with learnable frequencies and weights. **(d)** The resulting representation is optimized through its induced neural tangent kernel (NTK), whose spectrum provides a perspective on training dynamics.

in rotational flows are naturally coupled, while their interaction with time t can often be modeled through a lower-rank structure.

These observations suggest that effective PINN representations should adapt both *what frequencies are represented* and *how variables interact*. We therefore develop a unified representation framework with two complementary components.

First, we introduce *adaptive multi-frequency representations* (MFAR), in which each neural factor combines multiple sinusoidal branches with learnable frequencies and weights. MFAR provides a multi-frequency parameterization whose spectral content adapts to the target PDE during optimization, rather than relying on fixed frequencies or a single learnable frequency.

Second, we introduce *structured variable separation* (SVS) for spatiotemporal representation. Instead of enforcing dimension-wise separability, SVS groups strongly correlated spatial variables and separates them from the temporal variable, preserving unrestricted interactions within the spatial group while imposing low-rank coupling across groups. It therefore provides an intermediate structure between fully coupled networks and dimension-wise tensor decompositions. The grouping follows the physical coupling of the variables rather than a particular PDE instance; when the groups are singletons, SVS reduces to a standard separable (CP-type) factorization. Moreover, MFAR is independent of this structural choice and can be directly incorporated into existing CP, TT, and Tucker parameterizations.

Learning the frequency content also raises a natural question: does adaptive frequency learning merely improve expressiveness, or does it alter the optimization geometry of PINNs? To investigate this, we analyze the neural tangent kernel (NTK) induced by MFAR-SVS. We derive its closed-form expression and characterize its infinite-width and infinite-rank limits. For adaptive frequency learning, the NTK decomposes into a fixed-frequency component and a positive-semidefinite correction induced by the trainable frequencies and branch weights. Consequently, all ordered eigenvalues, including the minimum eigenvalue, are non-decreasing relative to the fixed-frequency kernel. We further track the NTK spectrum and learned frequencies during training, showing that adaptive frequencies progressively capture higher-frequency components and are accompanied by a larger minimum NTK eigenvalue.

We evaluate the proposed representation on Flow Mixing, two-scale planar transport, Gaussian-vortex transport, and 3D Helmholtz equations. We compare adaptive and fixed-frequency parameterizations across different matrix and tensor decompositions, and examine solution accuracy, convergence, frequency adaptation, and NTK spectra. The results demonstrate consistent benefits of adaptive frequency learning, particularly for multi-scale and strongly coupled PDEs.

Our main contributions are summarized as follows:

Adaptive multi-frequency representation. We introduce a learnable multi-frequency parameterization that combines multiple sinusoidal branches with trainable frequencies and weights, enabling adaptive spectral representations during PINN optimization.

Structured variable separation. We develop a spatiotemporal low-rank representation that preserves interactions within strongly coupled spatial variables while imposing low-rank spatiotemporal coupling, reducing to a standard separable factorization when the groups are singletons.

NTK analysis. We derive the closed-form NTK and characterize its infinite-width and infinite-rank limits, showing that adaptive frequency learning induces a positive-semidefinite correction and hence non-decreasing ordered eigenvalues, including the minimum eigenvalue.

Empirical evaluation. Experiments on four PDE benchmarks show the effectiveness of the proposed representation, particularly for multi-scale and strongly coupled problems.

The rest of this paper is organized as follows. Section 2 reviews related work; Section 3 introduces the MFAR-SVS architecture and its universal approximation property; Section 4 presents the NTK analysis and spectral theorems; Section 5 reports the experimental results; Section 6 discusses limitations and future work. Proofs and additional experimental details are provided in the appendices.

2 Related Work

2.1 Implicit neural representations and PINNs

Implicit neural representations (INRs) model continuous functions with neural networks (Mildenhall et al., 2020). Physics-informed neural networks (PINNs) (Raissi et al., 2019; Karniadakis et al., 2021) extend this paradigm to PDEs by enforcing governing equations through automatic differentiation (Baydin et al., 2018). However, coordinate-based networks exhibit spectral bias, tending to learn low-frequency components before high-frequency ones (Rahaman et al., 2019; Krishnapriyan et al., 2021; Wang et al., 2021a; Hao et al., 2023). Sinusoidal representations such as SIREN (Sitzmann et al., 2020) and Fourier features (Tancik et al., 2020; Liu et al., 2020) improve high-frequency representation, but their frequency bands are typically fixed or manually specified. Adaptive activations (Jagtap et al., 2020) and learnable-basis networks (Liu et al., 2025) provide additional flexibility, but generally do not explicitly parameterize multiple learnable sinusoidal frequencies and their weights within each activation. In contrast, MFAR jointly learns multiple frequencies and their weights within each activation.

2.2 Tensor decompositions and variable coupling in PINNs

Low-rank PINNs exploit structured factorization to reduce the computational cost of high-dimensional PDEs. Existing CP, tensor-train, and Tucker (Vemuri et al., 2024) representations, along with related separable formulations, parameterize solutions through lower-dimensional factor networks. While effective for nearly separable solutions, they typically impose dimension-wise separability, which can restrict interactions among strongly correlated variables. Our structured variable separation instead groups correlated variables within subnetworks and imposes low-rank coupling only across groups, providing an intermediate structure between fully coupled networks and dimension-wise tensor decompositions.

2.3 Neural tangent kernel theory

The neural tangent kernel (NTK) (Jacot et al., 2018; Lee et al., 2018; Du et al., 2019) provides a kernel-based perspective on neural network optimization and has been used to analyze spectral bias and training dynamics in PINNs (Wang et al., 2021b;a; 2022). In the wide-network regime, the NTK

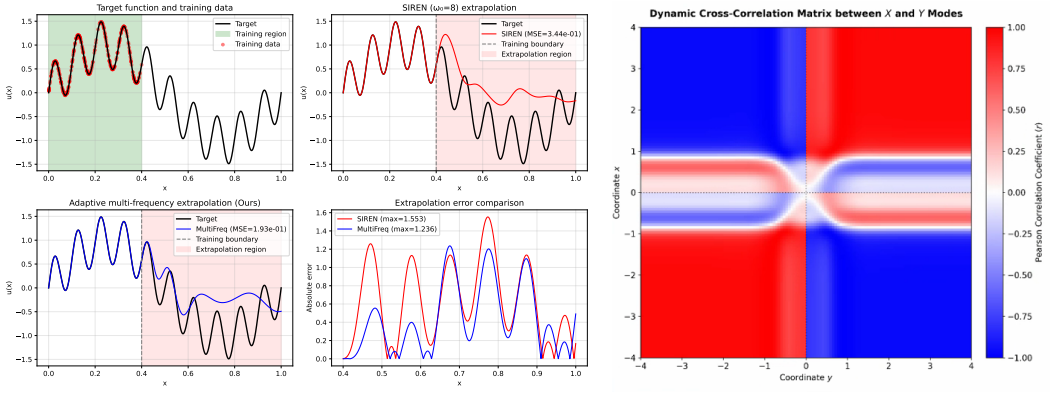


Figure 2: Motivations for adaptive multi-frequency representation and structured variable separation. **Left**: extrapolation of the multi-scale target $f(x) = \sin(2\pi x) + 0.5\cos(20\pi x)$ trained on $x \in [0, 0.4]$, comparing a fixed-frequency SIREN with MFAR. **Right**: cross-dimensional correlation of the Flow Mixing solution $u(T, X, Y)$, measured by the Pearson correlation between the X and Y modes across time, illustrating the strong dependence between the spatial variables.

spectrum characterizes the convergence rates of different function modes, with small eigenvalues corresponding to slowly learned components. Existing analyses mainly consider fixed parameterizations or use NTK insights to motivate loss reweighting and training strategies (Rathore et al., 2024). In contrast, we analyze how trainable frequency parameters alter the NTK itself, deriving a closed-form kernel for the proposed structured representation and establishing its resulting spectral properties.

3 Method

3.1 Adaptive Multi-Frequency Representation

Neural networks exhibit a spectral bias toward low-frequency components (Rahaman et al., 2019; Wang et al., 2021a). SIREN (Sitzmann et al., 2020) alleviates this issue using sinusoidal activations,

$$\phi(x) = \sin\big(\omega_0(Wx + b)\big), \tag{1}$$

where ω_0 is a ***fixed*** frequency scale. However, a single prescribed scale may be *insufficient* for PDE solutions containing multiple spatial or temporal frequencies.

We therefore introduce an adaptive multi-frequency activation with K parallel branches:

$$\mathbf{a}^{(l)} = \frac{1}{K}\sum_{k=1}^{K} c_k\, \sigma\Big(n_k(\mathbf{W}^\top \mathbf{a}^{(l-1)} + \mathbf{b})\Big)\,, \tag{2}$$

where $\sigma(\cdot) = \sin(\cdot)$, n_k is a trainable frequency scale, and c_k is a learnable branch weight. Thus, the representation can be viewed as

$$u_i(\boldsymbol{\xi}) \approx \frac{1}{K}\sum_{k} c_k\, \sigma\big(n_k(\mathbf{w}^\top \boldsymbol{\xi} + b)\big)\,. \tag{3}$$

The learnable frequency scales and branch weights allow MFAR to adapt the spectral composition of the neural representation. When $K = 1$ and n_1 is fixed, Eq. equation 2 reduces to SIREN; with a single trainable frequency branch, it recovers the adaptive activation of (Jagtap et al., 2020). Figure 2(left) compares a fixed-frequency SIREN with MFAR on the multi-scale target $f(x) = \sin(2\pi x) + 0.5\cos(20\pi x)$, illustrating the benefit of learning multiple frequency scales for extrapolation. In contrast, the Flow Mixing solution in Figure 2(right) exhibits strong dependence between the two spatial variables, which cannot be addressed by frequency adaptation alone and motivates the structured variable separation introduced next.

3.2 Structured Variable Separation

Dimension-wise tensor decompositions such as CP, Tucker, and TT (Figure 1) impose separability across individual variables. This is effective when the variables are weakly coupled, but can be restrictive for spatiotemporal solutions with strong cross-dimensional spatial dependence. As illustrated by the Flow Mixing solution in Figure 2(right), the spatial variables X and Y can exhibit substantial dependence.

We therefore introduce *structured variable separation* (SVS) to separate temporal and spatial variables while preserving strongly coupled spatial variables within the same group. Specifically, we partition $\mathbf{x} = (\boldsymbol{\xi}, \boldsymbol{\eta}), \boldsymbol{\xi} \in \Omega_\xi \subset \mathbb{R}^p, \boldsymbol{\eta} \in \Omega_\eta \subset \mathbb{R}^q, p+q = d$, and model the cross-group interaction by a low-rank separation,

$$u_{\mathrm{SVS}}(\boldsymbol{\xi}, \boldsymbol{\eta}; \Theta) = \frac{1}{\sqrt{R}} \sum_{i=1}^{R} u_i(\boldsymbol{\xi}) v_i(\boldsymbol{\eta}) = \frac{1}{\sqrt{R}} \mathbf{U}(\boldsymbol{\xi})^\top \mathbf{V}(\boldsymbol{\eta}), \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the outputs of two subnetworks and R is the separation rank. The separation class is dense in $C(\Omega_\xi \times \Omega_\eta)$ under the uniform norm (Light & Cheney, 1985), with the corresponding approximation result for the full architecture given in Theorem 1.

The subnetworks use MFAR in their hidden layers, giving

$$u_{\text{MFAR-SVS}}(\boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{1}{\sqrt{R}} \mathbf{U}_{\mathrm{MFAR}}(\boldsymbol{\xi})^\top \mathbf{V}_{\mathrm{MFAR}}(\boldsymbol{\eta}). \tag{5}$$

Thus, MFAR adapts the spectral representation within each variable group, while SVS models their cross-group interaction. *When spatial variables are weakly coupled, no grouping is needed and the model reduces to a standard separable (CP-type) factorization.*

For comparison, we use a unified naming convention for factorized PINNs. CP, Tucker, and TT models with fixed-frequency sinusoidal activations are denoted as CP/Tucker/TT-fixed, while replacing their factor activations with MFAR yields MFAR-CP/Tucker/TT. On all benchmarks, the fixed-frequency and adaptive variants use the same $K = 3$ frequency components, with the fixed variants keeping the corresponding frequencies frozen. Thus, each fixed/adaptive pair differs only in whether the frequencies are learnable.

3.3 Theoretical Expressiveness

The structured separation in Eq. equation 5 raises a natural question: *does the low-rank coupling between variable groups restrict the expressive power of the resulting representation?* The following theorem shows that, with sufficiently large separation rank and sub-network capacity, MFAR-SVS retains the universal approximation property.

Theorem 1 (Universal approximation of MFAR-SVS PINN)**.** *For any continuous function $f \in C(\Omega_\xi \times \Omega_\eta)$ and any $\epsilon > 0$, there exist sub-networks NN_U, NN_V and a finite rank R such that*

$$\sup_{(\boldsymbol{\xi}, \boldsymbol{\eta}) \in \Omega} \left| f(\boldsymbol{\xi}, \boldsymbol{\eta}) - R^{-1/2} \sum_{i=1}^{R} u_i(\boldsymbol{\xi}) v_i(\boldsymbol{\eta}) \right| < \epsilon. \tag{6}$$

The proof combines the Stone–Weierstrass theorem with the universal approximation property of MLPs (Leshno et al., 1993); see Appendix A. Thus, structured variable separation preserves universal approximation when the rank and sub-network capacity are sufficiently large. Approximation expressiveness, however, does not characterize training efficiency. The NTK analysis in Section 4 therefore studies the resulting optimization dynamics.

4 Convergence and NTK Analysis

We analyze the training dynamics of the MFAR-SVS PINN through its neural tangent kernel (NTK). Under gradient flow for the squared loss,

$$\frac{d\mathbf{u}_t}{dt} = -K_\Theta(\mathbf{u}_t - \mathbf{y}), \tag{7}$$

where K_Θ is the NTK evaluated on the training points. Its eigenvalues determine the convergence rates of the corresponding kernel modes; in particular, $\lambda^+_{\min}(K_\Theta)$ controls the slowest asymptotic rate, while the condition number characterizes the imbalance between modes.

4.1 NTK of Structured Variable Separation

For the SVS representation $f_\Theta(\boldsymbol{\xi}, \boldsymbol{\eta}) = \sum_{r=1}^{R} u_r(\boldsymbol{\xi}; \Theta_U) v_r(\boldsymbol{\eta}; \Theta_V)$, the disjoint parameterization of the two sub-networks yields the following kernel decomposition.

Lemma 2 (NTK of SVS)**.** *Let $f_\Theta(\boldsymbol{\xi}, \boldsymbol{\eta}) = R^{-1/2} \sum_{r=1}^{R} u_r(\boldsymbol{\xi}; \Theta_U)\, v_r(\boldsymbol{\eta}; \Theta_V)$ with disjoint parameter sets Θ_U, Θ_V, with the normalization of Eq. equation 4. Then the global NTK is*

$$K_\Theta(\mathbf{x}, \mathbf{x}') = \frac{1}{R}\Big(\mathbf{v}(\boldsymbol{\eta})^\top \mathbf{K}_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \mathbf{v}(\boldsymbol{\eta}') + \mathbf{u}(\boldsymbol{\xi})^\top \mathbf{K}_V(\boldsymbol{\eta}, \boldsymbol{\eta}')\, \mathbf{u}(\boldsymbol{\xi}')\Big), \tag{8}$$

where $(\mathbf{K}_U)_{rs} = \langle \nabla_{\Theta_U} u_r(\boldsymbol{\xi}), \nabla_{\Theta_U} u_s(\boldsymbol{\xi}') \rangle$ and $(\mathbf{K}_V)_{rs} = \langle \nabla_{\Theta_V} v_r(\boldsymbol{\eta}), \nabla_{\Theta_V} v_s(\boldsymbol{\eta}') \rangle$.

The cross term vanishes because Θ_U and Θ_V are disjoint. Thus, SVS induces an additive NTK structure associated with the two variable groups, providing a direct connection between structured separation and training dynamics.

4.2 Large-Width and Large-Rank Limit

The kernel decomposition in Lemma 2 also gives a deterministic limit in the large-width and large-rank regime.

Theorem 3 (Asymptotic NTK of SVS)**.** *Under independent Gaussian initialization, in the two-stage limit (width first, then rank), the NTK converges almost surely to*

$$K_\Theta(\mathbf{x}, \mathbf{x}') \xrightarrow{a.s.} k_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, c_V(\boldsymbol{\eta}, \boldsymbol{\eta}') + c_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, k_V(\boldsymbol{\eta}, \boldsymbol{\eta}'), \tag{9}$$

where k_U and k_V are the limiting NTKs (gradient kernels) of the two sub-networks, and c_U and c_V are the output covariance kernels of a single branch of each (Appendix A).

The theorem shows that the limiting kernel preserves the variable-group structure while coupling the two groups through their covariance kernels.

4.3 Adaptive Frequencies and NTK Spectrum

MFAR augments the fixed-frequency parameterization with trainable frequency scales and branch weights. Let $\Theta_{\text{adapt}} = (\Theta_{\text{fixed}}, \Theta_{\text{freq}})$, where Θ_{freq} contains the additional trainable frequency-related parameters. The corresponding NTK decomposes as

$$K^{\text{adapt}} = K^{\text{fixed}} + \Delta K,\ \Delta K = J_{\text{freq}} J_{\text{freq}}^\top \succeq 0, \tag{10}$$

where J_{freq} denotes the Jacobian with respect to Θ_{freq}.

Proposition 4 (Spectral improvement from adaptive frequencies)**.** *Let $K^{\text{adapt}} = K^{\text{fixed}} + \Delta K$, evaluated at the same parameter values, with $\Delta K \succeq 0$ and $K^{\text{fixed}} \succ 0$. Then for every eigenvalue index m (in increasing order),*

$$\lambda_m(K^{\text{adapt}}) \geq \lambda_m(K^{\text{fixed}}), \tag{11}$$

and in particular

$$\lambda^+_{\min}(K^{\text{adapt}}) \geq \lambda^+_{\min}(K^{\text{fixed}}). \tag{12}$$

The result follows from Weyl's monotonicity theorem. Hence, under the above parameter embedding, adaptive frequency parameters cannot decrease the NTK eigenvalues and cannot worsen the theoretical worst-case convergence rate under gradient flow. An improvement in the condition number is not implied by this result and is examined empirically.

5 Experiments

We evaluate MFAR-SVS PINN in terms of solution accuracy, NTK spectral properties, and training dynamics; component ablations are reported in Appendix B.4. All architectures, losses, and training protocols are implemented consistently in JAX/Flax. The NTK measurements, parameter counts, and ablation settings are reported in Appendix B.1, which also gives the complete experimental details.

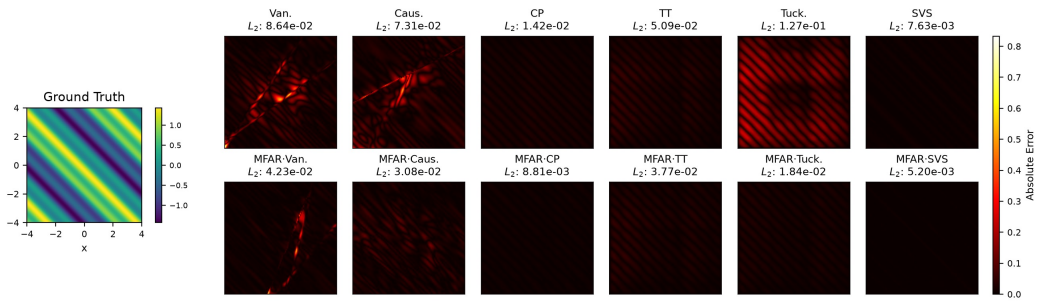
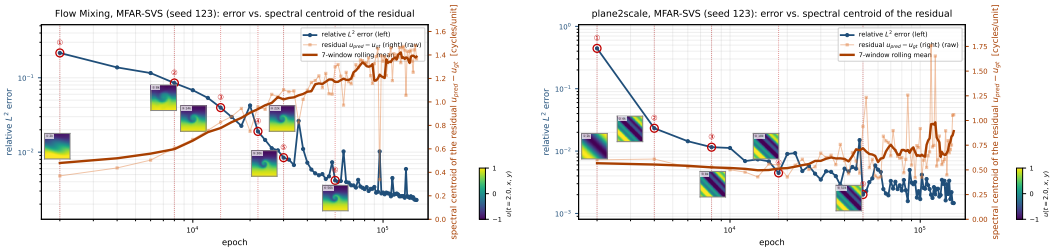


Figure 3: Two-scale planar transport: ground-truth solution and pointwise absolute errors at $t = 2.0$. **Top**: fixed-frequency models; **bottom**: MFAR variants. Relative L^2 errors are shown above each error map, all using the same color scale.



(a) Flow Mixing (b) two-scale planar transport

Figure 4: Training dynamics of MFAR-SVS on the Flow Mixing 1 (a) and two-scale planar transport 1 (b) benchmarks. Left axis: relative L^2 error; right axis: spectral centroid of the residual $u_{\text{pred}} - u_{\text{gt}}$, in cycles per unit length.

5.1 Experimental setup

Baselines. We compare with Vanilla-PINN (Raissi et al., 2019), Causal-PINN (Wang et al., 2024), CP-PINN, TT-PINN, and Tucker-PINN (Vemuri et al., 2024; Cho et al., 2023), together with fixed- and adaptive-frequency variants that isolate the effects of MFAR and SVS.

Implementation. All models are implemented in JAX/Flax and optimized with *Adam* (Kingma & Ba, 2015). * marks configurations that the non-decomposed arms cannot run at $N_C = 128$ (Appendix B.1). Here N_C denotes the number of collocation points per coordinate direction and R the rank of the respective factorization. Unless otherwise specified, $K = 3$ frequency components are used, while the fixed-frequency variants freeze the same $K = 3$ components. No paired solution data are used: collocation points and initial/boundary points are randomly sampled from their corresponding domains and surfaces. All methods optimize

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}}\mathcal{L}_{\text{IC}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}}, \tag{13}$$

where the three terms denote the PDE residual, initial-condition, and boundary-condition losses. Performance is measured by the relative L^2 error on a 128^3 test grid, averaged over three random seeds. The non-decomposed arms are infeasible at $N_C = 128$; even at $N_C = 64$ they take 596.6 s and 595.0 s to train on Flow Mixing, against 89.3–112.1 s for the fixed-frequency CP, TT, and Tucker models (Table 1).

5.2 Benchmark equations and numerical results

Benchmarks. Our four benchmarks cover radial dispersion, multi-scale structure, variable separation, and high-frequency oscillation, probing complementary failure modes of fixed-frequency PINNs: *Flow Mixing*, where a sharp interface undergoes radius-dependent rotation that cannot be resolved by a single carrier frequency; *two-scale planar transport*, with a multi-scale but weakly entangled target; *Gaussian-vortex transport*, which combines a fast-varying core with a nearly rigid exterior; and *3D Helmholtz*, whose separable, highly oscillatory rank-one solution isolates expressivity from variable coupling. Full formulations, domains, and initial conditions, together with those of Klein–Gordon and 5D Poisson, are given in Appendix B.2. Tables 1–4 report the errors of all compared methods.

Table 1: Relative L^2 errors on *Flow Mixing* across eight configurations, averaged over 3 seeds. Fixed/adaptive models use fixed/learnable frequencies; bold denotes the best result per row.

			Monolithic		Fixed frequency				Adaptive multi-frequency (MFAR)					
Setting	N_C	R	Van.	Caus.	CP	TT	Tuck.	SVS	Van.	Caus.	CP	TT	Tuck.	SVS
Flow Mixing 1	64	32	0.055	0.060	0.044	0.052	0.050	0.031	0.041	0.054	0.044	0.050	0.050	**0.017**
Flow Mixing 1	64	64	0.055	0.060	0.045	0.053	0.046	0.023	0.041	0.054	0.044	0.053	0.041	**0.020**
Flow Mixing 1	128	32	*	*	0.003	0.002	0.003	0.002	*	*	0.003	0.002	0.002	**0.002**
Flow Mixing 1	128	64	*	*	0.002	0.003	0.003	0.003	*	*	0.002	0.003	0.003	**0.002**
Mean runtime (s)			596.6	595.0	89.3	99.4	112.1	369.8	1893.0	1911.1	204.7	286.7	228.3	415.4
Flow Mixing 2	64	32	0.078	0.098	0.065	0.077	0.073	0.064	0.062	0.091	0.059	0.071	0.070	**0.035**
Flow Mixing 2	64	64	0.078	0.098	0.070	0.077	0.074	0.055	0.062	0.091	0.066	0.068	0.069	**0.045**
Flow Mixing 2	128	32	*	*	0.008	0.008	0.007	0.007	*	*	0.008	0.007	0.007	**0.006**
Flow Mixing 2	128	64	*	*	0.007	0.008	0.009	0.007	*	*	0.007	0.008	0.008	**0.006**
Mean runtime (s)			589.8	603.6	91.9	106.6	110.5	352.0	1895.2	1907.8	198.9	200.1	208.9	408.8

Table 2: Relative L^2 errors on *two-scale planar transport* across eight configurations, averaged over 3 seeds. Fixed/adaptive models use fixed/learnable frequencies; bold denotes the best result per row.

			Monolithic		Fixed frequency				Adaptive multi-frequency (MFAR)					
Setting	N_C	R	Van.	Caus.	CP	TT	Tuck.	SVS	Van.	Caus.	CP	TT	Tuck.	SVS
Plane 1	64	32	0.085	0.073	0.015	0.028	0.115	0.007	0.042	0.031	0.009	0.023	0.017	**0.004**
Plane 1	64	16	0.085	0.073	0.011	0.024	0.046	0.008	0.042	0.031	0.008	0.013	0.011	**0.005**
Plane 1	128	32	*	*	0.004	0.009	0.010	0.007	*	*	**0.003**	0.007	0.005	0.004
Plane 1	128	16	*	*	0.006	0.006	0.003	0.006	*	*	0.004	0.005	**0.003**	0.006
Mean runtime (s)			531.9	537.7	188.8	177.9	189.2	318.2	1697.2	1709.7	186.4	189.8	181.9	353.8
Plane 2	64	32	0.089	0.072	0.021	0.027	0.063	0.007	0.075	0.043	0.011	0.023	0.019	**0.005**
Plane 2	64	16	0.089	0.072	0.009	0.022	0.016	0.011	0.075	0.043	0.008	0.014	0.008	**0.004**
Plane 2	128	32	*	*	0.004	0.009	0.009	0.005	*	*	0.004	0.006	0.005	**0.004**
Plane 2	128	16	*	*	0.005	0.005	0.006	0.005	*	*	0.004	0.005	**0.003**	0.005
Mean runtime (s)			526.6	544.9	210.6	174.1	171.4	317.2	1829.9	1950.5	182.8	181.6	183.1	356.1

Results. Across the four benchmarks, learnable frequencies never worsen a decomposition (116 of 156 improve). Where variables are strongly coupled, SVS is most accurate—on Flow Mixing, Gaussian-vortex transport, and low-collocation planar transport (Figure 3). On the separable Helmholtz problem, dimension-wise factorizations are more accurate; SVS nonetheless stays 1.2–11$\times$ below the monolithic error and MFAR helps all five variants.

5.3 NTK and Training Dynamics

Training dynamics. Figure 4 tracks the relative L^2 error and the spectral content of the residual during training. The error decreases from 2.2×10^{-1} to 2.3×10^{-3}, while the 7-point rolling mean of the residual spectral centroid increases from 0.43 to 1.38 cycles per unit length. This shift indicates that the remaining error becomes progressively concentrated at higher frequencies, consistent with the role of adaptive frequency representations in resolving fine-scale components.

NTK spectrum. Figure 5 compares the NTK spectra of SIREN PINN (Sitzmann et al., 2020), SVS-PINN, and MFAR-SVS PINN on Flow Mixing using the same collocation points and initialization; the kernel measured here is the network-output NTK, evaluated on the training collocation set. Both structured variable separation and learnable frequencies increase the minimal positive eigenvalue $\lambda^+_{\min}$ and reduce the condition number κ. The increase in $\lambda^+_{\min}$ is consistent with the spectral improvement established by Proposition 4; in turn, lifting the bottom of the spectrum can reduce the condition number when the largest eigenvalues remain relatively stable. The resulting better-conditioned NTK facilitates more efficient learning of solution components that are poorly represented by fixed-frequency features.

Ablation studies. We ablate the choice of the number of frequency components K to examine how the frequency composition affects MFAR-SVS performance. Results are reported in Appendix B.4.

Table 3: Relative L^2 errors on *Gaussian-vortex transport* problem across eight configurations, averaged over 3 seeds. Fixed/adaptive models use fixed/learnable frequencies; bold denotes the best result per row.

			Monolithic		Fixed frequency				Adaptive multi-frequency (MFAR)					
Setting	N_C	R	Van.	Caus.	CP	TT	Tuck.	SVS	Van.	Caus.	CP	TT	Tuck.	SVS
Gaussian vortex 1	64	32	0.048	0.044	0.026	0.023	0.022	0.015	0.043	0.039	0.022	0.020	0.019	**0.010**
Gaussian vortex 1	64	16	0.048	0.044	0.037	0.022	0.040	0.014	0.043	0.039	0.036	0.017	0.036	**0.009**
Gaussian vortex 1	128	32	*	*	0.012	0.007	0.007	0.007	*	*	0.012	0.007	0.007	**0.005**
Gaussian vortex 1	128	16	*	*	0.020	0.010	0.010	0.005	*	*	0.019	0.009	0.010	**0.005**
Mean runtime (s)			586.1	589.1	160.9	170.5	175.2	362.8	1890.3	1909.7	188.7	200.1	200.8	404.8
Gaussian vortex 2	64	32	0.059	0.052	0.026	0.024	0.022	0.044	0.047	0.041	0.022	0.021	0.020	**0.011**
Gaussian vortex 2	64	16	0.059	0.052	0.037	0.022	0.039	0.042	0.047	0.041	0.037	0.017	0.036	**0.009**
Gaussian vortex 2	128	32	*	*	0.012	0.007	0.006	0.009	*	*	0.012	0.007	0.006	**0.005**
Gaussian vortex 2	128	16	*	*	0.019	0.010	0.010	0.009	*	*	0.019	0.010	0.010	**0.005**
Mean runtime (s)			586.0	587.4	152.7	172.4	189.2	354.2	1892.4	1911.3	215.4	239.9	240.7	429.0

Table 4: Relative L^2 errors on the *3D Helmholtz* equation across eight configurations, averaged over 3 seeds. Fixed and adaptive models use fixed and learnable frequencies, respectively. Bold indicates the best result. Causal-PINN is omitted as it requires a time axis.

			Monolithic	Fixed frequency				Adaptive multi-frequency (MFAR)				
Setting	N_C	R	Van.	CP	TT	Tuck.	SVS	Van.	CP	TT	Tuck.	SVS
Helmholtz 1	64	32	0.079	0.211	0.001	0.002	0.008	0.059	0.184	0.001	**0.001**	0.006
Helmholtz 1	64	16	0.079	0.002	0.002	0.002	0.007	0.059	0.002	0.002	**0.002**	0.007
Helmholtz 1	128	32	*	0.147	0.001	0.002	0.007	*	0.114	0.001	**0.001**	0.006
Helmholtz 1	128	16	*	0.002	0.002	0.002	0.008	*	0.002	0.002	**0.002**	0.006
Mean runtime (s)			822.4	70.1	75.1	78.3	78.7	2844.4	189.6	211.2	220.7	181.2
Helmholtz 2	64	32	0.097	0.196	0.012	0.007	0.082	0.082	0.104	0.009	**0.005**	0.064
Helmholtz 2	64	16	0.097	0.005	0.011	0.008	0.067	0.082	**0.005**	0.010	0.007	0.060
Helmholtz 2	128	32	*	0.100	0.010	0.009	0.062	*	0.087	0.007	**0.007**	0.044
Helmholtz 2	128	16	*	0.006	0.007	0.008	0.049	*	**0.005**	0.006	0.007	0.036
Mean runtime (s)			825.7	79.5	93.2	96.4	76.4	2912.9	441.4	292.1	283.3	207.1

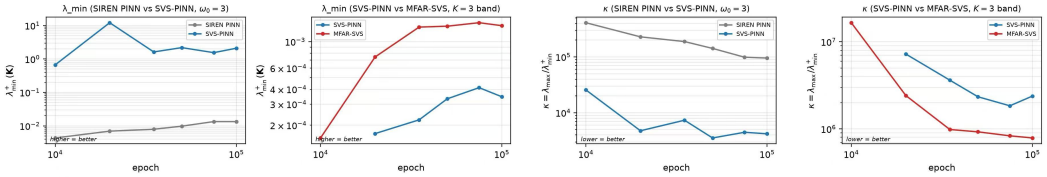


Figure 5: NTK spectra during training on Flow Mixing. **(a,c)**: SIREN PINN vs. SVS-PINN; **(b,d)**: SVS-PINN vs. MFAR-SVS PINN. **(a,b)** show $\lambda^{+}_{\min}$; **(c,d)** show the condition number κ. Only epochs $\geq 10^4$ are shown due to the float64 noise floor at earlier epochs.

6 Discussion and Conclusion

We presented MFAR-SVS PINN, which combines adaptive multi-frequency representation with structured variable separation. Across the four benchmarks, learnable frequencies never degrade a decomposition, and SVS is the most accurate model on the strongly coupled problems. The NTK analysis further shows that adaptive frequencies introduce a positive-semidefinite kernel component and therefore cannot decrease the ordered NTK eigenvalues, including $\lambda^{+}_{\min}$.

Limitations. Our NTK analysis assumes the wide-network regime and leaves trainable frequency dynamics uncharacterized. Increasing K adds computational cost with diminishing returns.

Future work. Promising directions include finite-width NTK analysis, adaptive rank selection, and transferring learned variable-group representations across related PDEs.

REFERENCES

A.G. Baydin, B.A. Pearlmutter, A.A. Radul, and J.M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.

Richard E Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 23761–23788, 2023. doi: 10.52202/075280-1032.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. doi: 10.1007/s10915-022-01939-z.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. arXiv preprint arXiv:2306.08827, 2023.

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

Ameya D Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.

George E Karniadakis, Ioannis G Kevrekidis, George Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/pdf?id=B1EA-M-0Z`.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. doi: 10.1016/S0893-6080(05)80131-5.

William Allan Light and Elliott Ward Cheney. *Approximation Theory in Tensor Product Spaces*, volume 1169 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1985. ISBN 978-3-540-16057-1. doi: 10.1007/BFb0075391.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, volume 2025, pp. 70367–70413, 2025.

Ziqi Liu, Wei Cai, and Zhi-Qin John Xu. Multi-scale deep neural network (mscalednn) for solving poisson-boltzmann equation in complex domains. *Communications in Computational Physics*, 28(5):1970–2001, 2020. doi: 10.4208/cicp.oa-2020-0179.

L. Lu, X. Meng, Z. Mao, and G.E. Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63:208–228, 2021.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pp. 405–421. Springer, 2020. doi: 10.1007/978-3-030-58452-8_24.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *International Conference on Machine Learning (ICML)*, 2024. doi: 10.48550/arXiv.2402.01868. arXiv:2402.01868.

V. Sitzmann, J.N.P. Martel, A.W. Bergman, D.B. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:7462–7473, 2020.

Gilbert Strang and George J Fix. *An Analysis of the Finite Element Method*. Wellesley-Cambridge Press, 2008.

John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Sai Karthikeya Vemuri, Tim Büchner, Julia Niebling, and Joachim Denzler. Functional tensor decompositions for physics-informed neural networks. In *International Conference on Pattern Recognition*, pp. 32–46. Springer, 2024.

S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43:A3055–A3081, 2021a.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021b. doi: 10.1016/j.cma.2021.113938.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022. doi: 10.1016/j.jcp.2021.110768.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024.

A PROOFS

Throughout this appendix every statement is restated in full before it is proved, so that the proofs can be read without consulting the main text; each restatement names the numbered statement of the main text it corresponds to, and every symbol is defined at its first use.

A.1 Universal approximation of MFAR-SVS

Theorem A.1 (Restatement of Theorem 1: Universal approximation of MFAR-SVS PINN)**.** *Let $\Omega_\xi \subset \mathbb{R}^p$ and $\Omega_\eta \subset \mathbb{R}^q$ be compact sets, $\Omega = \Omega_\xi \times \Omega_\eta$, and let $\|\cdot\|_\infty$ be the uniform norm on Ω, $\|g\|_\infty = \sup_{(\boldsymbol{\xi},\boldsymbol{\eta})\in\Omega} |g(\boldsymbol{\xi},\boldsymbol{\eta})|$. Let $NN_U : \Omega_\xi \to \mathbb{R}^R$ and $NN_V : \Omega_\eta \to \mathbb{R}^R$ be feed-forward sub-networks with R output coordinates each, and let $u_i : \Omega_\xi \to \mathbb{R}$ and $v_i : \Omega_\eta \to \mathbb{R}$ denote their i-th output coordinates ($i = 1, \dots, R$), so that the MFAR-SVS representation of separation rank R is $f_\Theta(\boldsymbol{\xi},\boldsymbol{\eta}) = \sum_{i=1}^R u_i(\boldsymbol{\xi})\, v_i(\boldsymbol{\eta})$. Then for every $f \in C(\Omega)$ and every $\epsilon > 0$ there exist a finite rank R and sub-networks NN_U, NN_V of rank R such that*

$$\sup_{(\boldsymbol{\xi},\boldsymbol{\eta})\in\Omega} \left| f(\boldsymbol{\xi},\boldsymbol{\eta}) - R^{-1/2} \sum_{i=1}^{R} u_i(\boldsymbol{\xi})\, v_i(\boldsymbol{\eta}) \right| < \epsilon. \tag{14}$$

Proof. The approximation is assembled in three steps: a continuous low-rank separated approximation of f, a neural realization of its factors, and an error budget that combines the two.

Step 1 (continuous separation of f). The collection of finite sums $\sum_i u_i(\boldsymbol{\xi})v_i(\boldsymbol{\eta})$ with continuous factors is an algebra of continuous functions on Ω: it is closed under addition and under multiplication (the product of two such sums expands into a finite sum of products, the scalar factors being absorbed into the functions), it contains the constants ($u_1 \equiv v_1 \equiv 1$), and it separates points of Ω: for $(\boldsymbol{\xi},\boldsymbol{\eta}) \neq (\boldsymbol{\xi}',\boldsymbol{\eta}')$ either $\boldsymbol{\xi} \neq \boldsymbol{\xi}'$, and the summand $\xi_j \cdot 1$ takes different values at the two points for some coordinate j, or $\boldsymbol{\eta} \neq \boldsymbol{\eta}'$, and the summand $1 \cdot \eta_k$ separates them. As Ω is compact and Hausdorff, the generalized Stone–Weierstrass theorem Light & Cheney (1985) makes this algebra dense in $C(\Omega)$ under $\|\cdot\|_\infty$. Hence there exist a rank $R^* \geq 1$ and continuous functions $u_i^* : \Omega_\xi \to \mathbb{R}$, $v_i^* : \Omega_\eta \to \mathbb{R}$ with

$$\left\| f - \sum_{i=1}^{R^*} u_i^*\, v_i^* \right\|_\infty < \frac{\epsilon}{2}. \tag{15}$$

Since Ω_ξ, Ω_η are compact, each u_i^*, v_i^* is bounded, so

$$M := \max_{1\leq i\leq R^*} \max\left\{ \|u_i^*\|_\infty, \|v_i^*\|_\infty \right\} \tag{16}$$

is finite. If $M = 0$ then all factors vanish, equation 15 gives $\|f\|_\infty < \epsilon/2$, and the claim holds by taking both sub-networks identically zero; we therefore assume $M > 0$ in the remainder.

Step 2 (neural realization of the factors). All R^* functions $\{u_i^*\}_{i=1}^{R^*}$ are realized by a single network with R^* output coordinates, and likewise $\{v_i^*\}_{i=1}^{R^*}$. The activation of the MFAR layer is $\sigma(x) = \sin(x)$, which is continuously differentiable and non-polynomial; universal approximation results for non-polynomial activations Leshno et al. (1993) therefore apply to $\sin$-activated multilayer perceptrons, and a network with several output coordinates approximates a vector-valued continuous map coordinate-wise. Moreover, the MFAR layer of Eq. (3) of the main text reduces to a standard sinusoidally activated layer upon setting $K = 1$, $n_1 = 1$, $c_1 = 1$, so the sub-networks considered here are at least as expressive as such perceptrons. Hence for every $\delta > 0$ there exist sub-networks NN_U, NN_V of rank R^* whose output coordinates satisfy

$$\|u_i^* - u_i\|_\infty < \delta, \qquad \|v_i^* - v_i\|_\infty < \delta, \qquad i = 1, \dots, R^*. \tag{17}$$

Step 3 (error budget). For each i, split the difference of products as

$$u_i^* v_i^* - u_i v_i = u_i^*\,(v_i^* - v_i) + v_i\,(u_i^* - u_i).$$

Here $\|u_i^*\|_\infty \leq M$ by equation 16, and $\|v_i\|_\infty \leq \|v_i^*\|_\infty + \|v_i^* - v_i\|_\infty \leq M + \delta$ by equation 17; using equation 17 once more,

$$\|u_i^* v_i^* - u_i v_i\|_\infty \leq M\delta + (M + \delta)\delta.$$

Summing this bound over the R^* components, the triangle inequality and equation 15 give

$$\left\| f - \sum_{i=1}^{R^*} u_i v_i \right\|_\infty \leq \frac{\epsilon}{2} + R^* M\delta + R^*(M + \delta)\delta. \tag{18}$$

Choose any δ with

$$\delta \le \min\Big\{\frac{\epsilon}{6R^*M},\, M\Big\}, \tag{19}$$

which is positive because $M > 0$ and $R^* \ge 1$. Then $R^*M\delta \le \epsilon/6$, while $\delta \le M$ yields $R^*(M+\delta)\delta \le 2R^*M\delta \le \epsilon/3$. Substituting both bounds into equation 18,

$$\Big\| f - \sum_{i=1}^{R^*} u_i v_i \Big\|_\infty < \frac{\epsilon}{2} + \frac{\epsilon}{6} + \frac{\epsilon}{3} = \epsilon,$$

which is equation 14 with $R = R^*$ up to the factor $R^{-1/2}$. That factor is absorbed into the output coordinates of NN_U, which may be rescaled by $R^{*1/2}$ without changing the architecture, so the bound is equivalent to the normalized form equation 14 stated above, proving the theorem. □

Remark. *The argument uses only two ingredients: density of finite separated sums in $C(\Omega)$ (Step 1) and the universal approximation property of the sub-networks together with their stability in the uniform norm (Steps 2–3). Neither the rank R^* nor the network capacity is bounded a priori; both grow with the accuracy ϵ and with the difficulty of f. Step 2 uses the fact that the reduction $K = 1$, $n_1 = 1$, $c_1 = 1$ turns an MFAR layer into a sinusoidally activated layer, which is what allows* sin*-activated sub-networks to be used as the factor approximators.*

A.2 NTK OF STRUCTURED VARIABLE SEPARATION

Lemma A.2 (Restatement of Lemma 2: NTK of SVS)**.** *Let the SVS representation be*

$$f_\Theta(\boldsymbol{\xi}, \boldsymbol{\eta}) = R^{-1/2} \sum_{r=1}^{R} u_r(\boldsymbol{\xi}; \Theta_U)\, v_r(\boldsymbol{\eta}; \Theta_V),$$

with the $R^{-1/2}$ normalization of Eq. (4) of the main text, where $\Theta = (\Theta_U, \Theta_V)$ is the disjoint union of the parameters of the two sub-networks, $\Theta_U \cap \Theta_V = \varnothing$; thus u_r depends on $\boldsymbol{\xi}$ and Θ_U only, and v_r on $\boldsymbol{\eta}$ and Θ_V only. For $\mathbf{x} = (\boldsymbol{\xi}, \boldsymbol{\eta})$ and $\mathbf{x}' = (\boldsymbol{\xi}', \boldsymbol{\eta}')$, define the neural tangent kernel of f_Θ by

$$K_\Theta(\mathbf{x}, \mathbf{x}') = \big\langle \nabla_\Theta f_\Theta(\mathbf{x}), \nabla_\Theta f_\Theta(\mathbf{x}') \big\rangle, \tag{20}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product of the parameter space. Let

$$\mathbf{v}(\boldsymbol{\eta}) = \big(v_1(\boldsymbol{\eta}), \dots, v_R(\boldsymbol{\eta})\big)^\top, \qquad \mathbf{u}(\boldsymbol{\xi}) = \big(u_1(\boldsymbol{\xi}), \dots, u_R(\boldsymbol{\xi})\big)^\top$$

collect the branch outputs, and let $\mathbf{K}_U, \mathbf{K}_V \in \mathbb{R}^{R\times R}$ be the sub-network kernels with entries

$$(\mathbf{K}_U)_{rs} = \big\langle \nabla_{\Theta_U} u_r(\boldsymbol{\xi}), \nabla_{\Theta_U} u_s(\boldsymbol{\xi}') \big\rangle, \qquad (\mathbf{K}_V)_{rs} = \big\langle \nabla_{\Theta_V} v_r(\boldsymbol{\eta}), \nabla_{\Theta_V} v_s(\boldsymbol{\eta}') \big\rangle.$$

Then

$$K_\Theta(\mathbf{x}, \mathbf{x}') = \frac{1}{R}\Big(\mathbf{v}(\boldsymbol{\eta})^\top \mathbf{K}_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \mathbf{v}(\boldsymbol{\eta}') + \mathbf{u}(\boldsymbol{\xi})^\top \mathbf{K}_V(\boldsymbol{\eta}, \boldsymbol{\eta}')\, \mathbf{u}(\boldsymbol{\xi}') \Big). \tag{21}$$

Proof. Because Θ_U and Θ_V are disjoint, the gradient of f_Θ splits as $\nabla_\Theta f_\Theta = (\nabla_{\Theta_U} f_\Theta, \nabla_{\Theta_V} f_\Theta)$ and the inner product equation 20 splits into the two blocks

$$K_\Theta = \big\langle \nabla_{\Theta_U} f_\Theta, \nabla_{\Theta_U} f_\Theta \big\rangle + \big\langle \nabla_{\Theta_V} f_\Theta, \nabla_{\Theta_V} f_\Theta \big\rangle,$$

with no cross term, since differentiating with respect to Θ_U never involves Θ_V.

For the Θ_U block, $v_r(\boldsymbol{\eta})$ does not depend on Θ_U, so it enters as a scalar coefficient:

$$\nabla_{\Theta_U} f_\Theta = R^{-1/2} \sum_{r=1}^{R} v_r(\boldsymbol{\eta})\, \nabla_{\Theta_U} u_r(\boldsymbol{\xi}).$$

Taking the inner product with the analogous expression at $\mathbf{x}'$ and using bilinearity,

$$\begin{aligned} \big\langle \nabla_{\Theta_U} f_\Theta, \nabla_{\Theta_U} f_\Theta \big\rangle &= R^{-1} \sum_{r,s=1}^{R} v_r(\boldsymbol{\eta}) v_s(\boldsymbol{\eta}') \big\langle \nabla_{\Theta_U} u_r(\boldsymbol{\xi}), \nabla_{\Theta_U} u_s(\boldsymbol{\xi}') \big\rangle \\ &= R^{-1} \mathbf{v}(\boldsymbol{\eta})^\top \mathbf{K}_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, \mathbf{v}(\boldsymbol{\eta}'), \end{aligned}$$

which is the first term of equation 21. The Θ_V block is symmetric, with $u_r(\boldsymbol{\xi})$ as the scalar coefficient, and gives the second term. Adding the two blocks yields equation 21, that is, Eq. (8) of the main text. □

A.3 WIDTH LIMIT AT FIXED SEPARATION RANK

Lemma A.3 (Width limit at fixed separation rank)**.** *Consider a single branch of the $\boldsymbol{\xi}$-sub-network,*

$$u(\boldsymbol{\xi}) = \frac{1}{\sqrt{W}} \sum_{i=1}^{W} (\mathbf{W}_2)_i \, \sigma\big(\mathbf{w}_i^\top \boldsymbol{\xi} + b_i\big), \tag{22}$$

where $\mathbf{w}_i \in \mathbb{R}^p$, $b_i \in \mathbb{R}$ and $(\mathbf{W}_2)_i \in \mathbb{R}$ denote the incoming weights, the bias and the outgoing weight of hidden unit i, $\sigma(x) = \sin(x)$ is the periodic activation with derivative $\dot\sigma(x) = \cos(x)$, and the parameters are initialized independently as $\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{I}_p)$, $b_i \sim \mathcal{N}(0,1)$ and $(\mathbf{W}_2)_i \sim \mathcal{N}(0,1)$. Write $z_i = \mathbf{w}_i^\top \boldsymbol{\xi} + b_i$ for the pre-activation of unit i, and let z, z' denote the quantities $\mathbf{w}^\top \boldsymbol{\xi} + b$ and $\mathbf{w}^\top \boldsymbol{\xi}' + b$ evaluated at one common draw of $(\mathbf{w}, b)$.

Then the NTK of a single branch converges in probability, as $W \to \infty$, to the deterministic limit

$$k_U(\boldsymbol{\xi}, \boldsymbol{\xi}') = \mathbb{E}\big[\sigma(z)\,\sigma(z')\big] + (\boldsymbol{\xi}^\top \boldsymbol{\xi}' + 1)\, \mathbb{E}\big[\dot\sigma(z)\,\dot\sigma(z')\big], \tag{23}$$

where the expectation is over $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_p)$, $b \sim \mathcal{N}(0,1)$; the kernel k_V of the $\boldsymbol{\eta}$-sub-network is defined identically with $\boldsymbol{\eta}$ in place of $\boldsymbol{\xi}$. Two branches with independently initialized parameters have NTK entries converging in probability to 0, so that the sub-network kernels converge to $k_U \mathbf{I}_R$ and $k_V \mathbf{I}_R$.

Write the SVS representation with the $R^{-1/2}$ normalization of the main text, $f_\Theta(\boldsymbol{\xi}, \boldsymbol{\eta}) = R^{-1/2} \sum_{r=1}^{R} u_r(\boldsymbol{\xi}) v_r(\boldsymbol{\eta})$, so that the NTK identity equation 21 of Lemma A.2 carries the prefactor R^{-1}; without this rescaling the kernel would scale linearly with R. Consequently, if the separation rank R is held fixed and the two sub-networks are initialized independently as above, the NTK of the SVS representation converges in distribution to

$$K_\Theta(\mathbf{x}, \mathbf{x}') \xrightarrow{\;d\;} k_U(\boldsymbol{\xi}, \boldsymbol{\xi}')\, V_V(\boldsymbol{\eta}, \boldsymbol{\eta}') + k_V(\boldsymbol{\eta}, \boldsymbol{\eta}')\, V_U(\boldsymbol{\xi}, \boldsymbol{\xi}'), \tag{24}$$

where

$$V_V(\boldsymbol{\eta}, \boldsymbol{\eta}') = \frac{1}{R} \sum_{r=1}^{R} v_r(\boldsymbol{\eta})\, v_r(\boldsymbol{\eta}'), \qquad V_U(\boldsymbol{\xi}, \boldsymbol{\xi}') = \frac{1}{R} \sum_{r=1}^{R} u_r(\boldsymbol{\xi})\, u_r(\boldsymbol{\xi}') \tag{25}$$

are the empirical covariances of the branch outputs of the width limit. The convergence is in distribution and not in probability, because the sub-network kernels concentrate while the branch outputs do not; and because R is not sent to infinity, V_U and V_V do not converge to deterministic limits: the kernel equation 24 is itself random.

Proof. Throughout, the NTK of one branch is $K^{(W)} = \langle \nabla u(\boldsymbol{\xi}), \nabla u(\boldsymbol{\xi}') \rangle$, the gradient being taken over the branch parameters $\{\mathbf{w}_i, b_i, (\mathbf{W}_2)_i\}_{i=1}^W$; differentiating equation 22,

$$\frac{\partial u}{\partial (\mathbf{W}_2)_i} = \frac{1}{\sqrt{W}} \sigma(z_i), \qquad \frac{\partial u}{\partial \mathbf{w}_i} = \frac{1}{\sqrt{W}} (\mathbf{W}_2)_i \, \dot\sigma(z_i)\, \boldsymbol{\xi}, \qquad \frac{\partial u}{\partial b_i} = \frac{1}{\sqrt{W}} (\mathbf{W}_2)_i \, \dot\sigma(z_i). \tag{26}$$

Step 1 (single-branch limit). Hidden units have independent parameters, so only the contributions of the same unit i pair up, and equation 26 gives

$$K^{(W)} = \frac{1}{W} \sum_{i=1}^{W} \Big[\sigma(z_i)\sigma(z_i') + (\mathbf{W}_2)_i^2 \, \dot\sigma(z_i)\dot\sigma(z_i') \, \big(\boldsymbol{\xi}^\top \boldsymbol{\xi}' + 1\big) \Big],$$

the first term collecting the outgoing-weight contributions and the second those of the incoming weights ($\boldsymbol{\xi}^\top \boldsymbol{\xi}'$) and of the biases (1). The summands are identically distributed and independent across i, and are integrable because $|\sigma| \le 1$ and $|\dot\sigma| \le 1$ while $\mathbb{E}[(\mathbf{W}_2)_i^2] = 1$. Since the summands are also independent of $(\mathbf{W}_2)_i$ in the first term, the weak law of large numbers gives

$$K^{(W)} \xrightarrow{\;\mathbb{P}\;} \mathbb{E}\big[\sigma(z)\sigma(z')\big] + (\boldsymbol{\xi}^\top \boldsymbol{\xi}' + 1)\, \mathbb{E}\big[\dot\sigma(z)\dot\sigma(z')\big] = k_U(\boldsymbol{\xi}, \boldsymbol{\xi}'),$$

which is equation 23.

Step 2 (vanishing cross-branch entries). The R output coordinates share the hidden trunk $(\mathbf{w}_i, b_i)_{i=1}^W$ and differ only in their outgoing weights $(\mathbf{W}_2)_{ji}$, which is the architecture of the main

text. Let u_j and $u_{j'}$, $j \neq j'$, be two of them. Pairing the shared hidden units in the definition $(\mathbf{K}_U)_{jj'} = \langle \nabla_{\Theta_U} u_j(\boldsymbol{\xi}), \nabla_{\Theta_U} u_{j'}(\boldsymbol{\xi}') \rangle$,

$$K_{jj'}^{(W)} = \frac{1}{W} \sum_{i=1}^{W} (\mathbf{W}_2)_{ji} (\mathbf{W}_2)_{j'i} \, \dot{\sigma}(z_i) \dot{\sigma}(z_i') \big(\boldsymbol{\xi}^\top \boldsymbol{\xi}' + 1 \big),$$

where the outgoing-weight channel is absent because the two branch derivatives are supported on different outgoing weights ($\partial u_j / \partial (\mathbf{W}_2)_{j''i} = \delta_{jj''} W^{-1/2} \sigma(z_i)$ and likewise for j', with $j \neq j'$), while $z_i = z_i' = \mathbf{w}_i^\top \boldsymbol{\xi} + b_i$ is common to both branches. Every summand is a product of two functions of mutually independent parameters, so its expectation factorizes as $\frac{1}{W} \mathbb{E}[(\mathbf{W}_2)_j] \mathbb{E}[(\mathbf{W}_2)_{j'}] (\boldsymbol{\xi}^\top \boldsymbol{\xi}' + 1) \mathbb{E}[\dot{\sigma}(z)] \mathbb{E}[\dot{\sigma}(z')] = 0$, because the outgoing weights are centered. The summands are bounded and independent across i, hence $\mathrm{Var}\big(K_{jj'}^{(W)}\big) = O(1/W)$, and Chebyshev's inequality yields $K_{jj'}^{(W)} \xrightarrow{\mathbb{P}} 0$. As R is fixed, only finitely many pairs (j, j') occur, so the sub-network kernel converges to $k_U \mathbf{I}_R$; the $\boldsymbol{\eta}$-side is identical and gives $k_V \mathbf{I}_R$.

Step 3 (assembly). Substituting the two limits into the NTK identity equation 21 of Lemma A.2, whose prefactor R^{-1} cancels the R-fold sum in the empirical covariances equation 25, gives

$$K_\Theta(\mathbf{x}, \mathbf{x}') \xrightarrow{d} k_U(\boldsymbol{\xi}, \boldsymbol{\xi}') \, \frac{1}{R} \sum_{r=1}^{R} v_r(\boldsymbol{\eta}) v_r(\boldsymbol{\eta}') + k_V(\boldsymbol{\eta}, \boldsymbol{\eta}') \, \frac{1}{R} \sum_{r=1}^{R} u_r(\boldsymbol{\xi}) u_r(\boldsymbol{\xi}'),$$

which is equation 24 with the covariances equation 25, the branch outputs being those of the width limit.

The mode of convergence deserves a word, since the two ingredients do not converge in the same sense. The sub-network kernel $\mathbf{K}_U^{(W)}$ converges *in probability* to the constant $k_U \mathbf{I}_R$ (Steps 1–2), but the branch outputs do not concentrate: their fluctuations survive the width limit as a non-degenerate Gaussian process Lee et al. (2018). The two blocks depend on disjoint parameter sets and are therefore independent, so the product falls under Slutsky's theorem — a factor converging in probability to a constant multiplied by a factor converging in distribution — which delivers convergence *in distribution* of the assembled kernel, and nothing stronger. The empirical covariances equation 25 are averages of R branch outputs of the width limit, and they remain random because R is fixed. □

Remark. *The reason for isolating this step is that it separates the two limits involved in the asymptotic statement of Theorem 3: the width limit produces the deterministic factors k_U, k_V together with empirical covariances that are still random, and only the subsequent rank limit turns those covariances into the deterministic kernels c_U, c_V.*

A.4 Asymptotic NTK of SVS

Theorem A.4 (Restatement of Theorem 3: Asymptotic NTK of SVS)**.** *Let f_Θ be the SVS representation of separation rank R built from two independently initialized Gaussian sub-networks, NN_U acting on $\boldsymbol{\xi} \in \Omega_\xi$ and NN_V acting on $\boldsymbol{\eta} \in \Omega_\eta$, and let K_Θ be its NTK equation 21. Under independent Gaussian initialization, in the two-stage limit (width first, then rank), the NTK converges almost surely to*

$$K_\Theta(\mathbf{x}, \mathbf{x}') \xrightarrow{a.s.} k_U(\boldsymbol{\xi}, \boldsymbol{\xi}') \, c_V(\boldsymbol{\eta}, \boldsymbol{\eta}') + c_U(\boldsymbol{\xi}, \boldsymbol{\xi}') \, k_V(\boldsymbol{\eta}, \boldsymbol{\eta}'), \tag{27}$$

where k_U and k_V are the limiting NTKs equation 23 of the two sub-networks, and c_U and c_V are their corresponding covariance kernels,

$$c_U(\boldsymbol{\xi}, \boldsymbol{\xi}') = \mathbb{E}\big[u(\boldsymbol{\xi}) u(\boldsymbol{\xi}')\big], \qquad c_V(\boldsymbol{\eta}, \boldsymbol{\eta}') = \mathbb{E}\big[v(\boldsymbol{\eta}) v(\boldsymbol{\eta}')\big],$$

the expectation being taken over the initialization of a single branch of the corresponding sub-network.

Proof. Following Lemma A.3 and its remark, the limit is taken in two stages: first the width at fixed rank, then the rank.

Step 1 (fixed rank). Let R be fixed. By Lemma A.3 the sub-network kernels converge in probability to $k_U \mathbf{I}_R$ and $k_V \mathbf{I}_R$, and the NTK of the representation converges in distribution to the kernel equation 24, whose randomness sits entirely in the empirical covariances V_V, V_U of equation 25.

Step 2 (rank limit). The R output coordinates of each sub-network share a common hidden trunk and are therefore *not* independent; the averaging over r in equation 25 is controlled in two stages, first over the trunk and then over the output layer.
(i) Trunk limit. Let H be the width of the shared trunk and $\mathbf{h}(\boldsymbol{\eta}) \in \mathbb{R}^H$ its feature vector, so that the branch output is $v_r(\boldsymbol{\eta}) = \mathbf{w}_r^\top \mathbf{h}(\boldsymbol{\eta})$ with i.i.d. output weights. The hidden units are i.i.d. across i and a single unit contributes an integrable random variable, so the weak law of large numbers gives $H^{-1}\mathbf{h}(\boldsymbol{\eta})^\top \mathbf{h}(\boldsymbol{\eta}') \xrightarrow{\mathbb{P}} \kappa_V(\boldsymbol{\eta}, \boldsymbol{\eta}') := \mathbb{E}[\sigma(z)\sigma(z')]$ as $H \to \infty$, the deterministic hidden-layer kernel of the neural-network Gaussian process limit Lee et al. (2018); the $\boldsymbol{\xi}$-side gives κ_U identically.
(ii) Output limit. Conditional on the trunk, the output weights are independent and identically distributed, and the summands of equation 25 have finite second moments because c_U and c_V are continuous on the compact domains; the conditional strong law of large numbers over r therefore gives $V_V(\boldsymbol{\eta}, \boldsymbol{\eta}') \to H^{-1}\mathbf{h}(\boldsymbol{\eta})^\top \mathbf{h}(\boldsymbol{\eta}')$ almost surely, which converges to $\kappa_V = c_V$ by (i). Combining (i) and (ii) and fixing a pair of evaluation points, the two-stage limit gives

$$V_V(\boldsymbol{\eta}, \boldsymbol{\eta}') \xrightarrow{\text{a.s.}} c_V(\boldsymbol{\eta}, \boldsymbol{\eta}'), \qquad V_U(\boldsymbol{\xi}, \boldsymbol{\xi}') \xrightarrow{\text{a.s.}} c_U(\boldsymbol{\xi}, \boldsymbol{\xi}').$$

Because the domains are separable and the covariance kernels are continuous, the convergence can be taken simultaneous on a countable dense subset and extended to all pairs by continuity.

Step 3 (assembly). Substituting Step 2 into equation 24 and using continuity of the sum and of the product, the random empirical covariances are replaced by c_V and c_U while the deterministic factors k_U, k_V are unaffected, which yields equation 27. □

Remark (Mode of convergence)**.** *The almost-sure statement is the one made in the main text. Read as the two-stage limit above, the rank limit is almost sure, through the strong law of large numbers applied to the branch outputs, whereas the width limit at fixed rank is* in distribution*: the sub-network kernels concentrate on deterministic limits, but the branch outputs do not, and their non-degenerate fluctuations are exactly what makes the fixed-rank kernel equation 24 random. The two stages are therefore stated separately, and no interchange of the limits is claimed. The in-distribution statement of the first stage can nevertheless be represented as an almost-sure convergence on a suitably coupled probability space, by Skorokhod's representation theorem; that is the sense in which the two-stage limit is to be read when the statement of Theorem 3 is taken at face value.*

A.5 Term-by-term expansion of $K^{\text{adapt}} = K^{\text{fixed}} + \Delta K$

Lemma A.5 (Term-by-term expansion of the adaptive NTK)**.** *Let u be the output of an MFAR sub-network with parameter vector $\theta = (\theta_{\text{base}}, \theta_{\text{freq}})$, where θ_{base} collects the linear weights and biases of all layers (the parameters that the fixed-frequency network also possesses) and $\theta_{\text{freq}} = \{n_k, c_k\}_{k=1}^K$ collects the learnable frequency scales $\{n_k\}_{k=1}^K$ and branch weights $\{c_k\}_{k=1}^K$ of the MFAR modules; the two blocks are disjoint, $\theta_{\text{base}} \cap \theta_{\text{freq}} = \varnothing$. For evaluation points $\mathbf{x}, \mathbf{x}'$ define*

$$K_\Theta(\mathbf{x}, \mathbf{x}') = \big\langle \nabla_\theta u(\mathbf{x}), \nabla_\theta u(\mathbf{x}') \big\rangle, \qquad K^{\text{fixed}}(\mathbf{x}, \mathbf{x}') = \big\langle \nabla_{\theta_{\text{base}}} u(\mathbf{x}), \nabla_{\theta_{\text{base}}} u(\mathbf{x}') \big\rangle.$$

Then

$$K_\Theta(\mathbf{x}, \mathbf{x}') = K^{\text{fixed}}(\mathbf{x}, \mathbf{x}') + \Delta K(\mathbf{x}, \mathbf{x}'),$$
$$\Delta K = \sum_{p \in \theta_{\text{freq}}} \nabla_p u(\mathbf{x})\, \nabla_p u(\mathbf{x}'). \tag{28}$$

where K^{fixed} is exactly the NTK of the network whose frequency block is frozen at the same parameter values, i.e. the fixed-frequency counterpart with identical base parameters. Consequently, on any finite set of collocation points the matrix ΔK is a sum of rank-one positive semi-definite matrices and hence $\Delta K \succeq 0$.

Proof. Since θ_{base} and θ_{freq} are disjoint, the parameter-space inner product splits into the two blocks without a cross term,

$$\begin{aligned} K_\Theta &= \nabla_{\theta_{\text{base}}} u(\mathbf{x}) \cdot \nabla_{\theta_{\text{base}}} u(\mathbf{x}') + \nabla_{\theta_{\text{freq}}} u(\mathbf{x}) \cdot \nabla_{\theta_{\text{freq}}} u(\mathbf{x}') \\ &= K^{\text{fixed}}(\mathbf{x}, \mathbf{x}') + \sum_{p \in \theta_{\text{freq}}} \nabla_p u(\mathbf{x})\, \nabla_p u(\mathbf{x}'), \end{aligned}$$

which is equation 28. The first term involves θ_{base} only, and freezing the frequencies leaves both the base block and its gradients unchanged, so this term is the NTK of the fixed-frequency counterpart with identical base parameters. For a collocation set $\{\mathbf{x}_m\}_{m=1}^N$, the summand indexed by p has entries $\nabla_p u(\mathbf{x}_m)\, \nabla_p u(\mathbf{x}_{m'})$, i.e. it is the rank-one matrix $\mathbf{g}_p \mathbf{g}_p^\top$ with $\mathbf{g}_p = \big(\nabla_p u(\mathbf{x}_1), \dots, \nabla_p u(\mathbf{x}_N)\big)^\top$. Such a matrix is positive semi-definite, because $\mathbf{a}^\top \mathbf{g}_p \mathbf{g}_p^\top \mathbf{a} = (\mathbf{g}_p^\top \mathbf{a})^2 \geq 0$ for every $\mathbf{a}$, and a sum of positive semi-definite matrices is positive semi-definite, so $\Delta K \succeq 0$. □

A.6 Spectral improvement from adaptive frequencies

Proposition A.6 (Restatement of Proposition 4: Spectral improvement from adaptive frequencies)**.** *Let $K^{\text{adapt}} = K^{\text{fixed}} + \Delta K$, where K^{fixed} and ΔK are the $N \times N$ NTK matrices of the frozen-frequency and adaptive networks evaluated on the same N collocation points, both at the same parameter values, and $\Delta K \succeq 0$ as established in Lemma A.5. We assume $K^{\text{fixed}} \succ 0$, so that $\lambda_{\min}^+ = \lambda_1$ for both kernels. Order the eigenvalues of a symmetric matrix increasingly, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. Then*

$$\lambda_m(K^{\text{adapt}}) \geq \lambda_m(K^{\text{fixed}}) \qquad \text{for every } m = 1, \dots, N, \tag{29}$$

and in particular

$$\lambda_{\min}^+(K^{\text{adapt}}) \geq \lambda_{\min}^+(K^{\text{fixed}}). \tag{30}$$

Proof. We first identify the parameter blocks and their gradients, then assemble the kernel, and finally read off its spectrum.

Step 1 (parameter blocks and gradients). Write one MFAR sub-network as

$$u(\boldsymbol{\xi}) = \frac{1}{\sqrt{W}} \sum_{i=1}^{W} (\mathbf{W}_2)_i \sum_{k=1}^{K} c_k\, \sigma\big(n_k\, z_i(\boldsymbol{\xi})\big), \qquad z_i(\boldsymbol{\xi}) = \mathbf{w}_i^\top \boldsymbol{\xi} + b_i, \tag{31}$$

where $\sigma(x) = \sin(x)$ is the periodic activation of Eq. (1) of the main text, $\dot\sigma$ its derivative, W the number of hidden units, $\{n_k\}_{k=1}^K$ the trainable frequencies, $\{c_k\}_{k=1}^K$ the branch weights, and $(\mathbf{W}_2)_i$, $\mathbf{w}_i$, b_i the outgoing weight, the incoming weights and the bias of hidden unit i; the MFAR activation applied here is that of Eq. (3) of the main text: all K branches share the linear map $(\mathbf{w}, \mathbf{b})$ of a hidden unit and differ only in their frequency scales n_k. The parameters separate into the base block $\theta_{\text{base}} = \big\{\mathbf{W}_2, \{\mathbf{w}_i, b_i\}_{i=1}^W\big\}$, which the fixed-frequency network also possesses, and the frequency block $\theta_{\text{freq}} = \{n_k, c_k\}_{k=1}^K$. With $z_i = z_i(\boldsymbol{\xi})$, $z_i' = z_i(\boldsymbol{\xi}')$ and $\dot\sigma$ the derivative of σ, the gradients of u are

$$\frac{\partial u}{\partial (\mathbf{W}_2)_i} = \frac{1}{\sqrt{W}} \sum_{k=1}^{K} c_k\, \sigma(n_k z_i), \qquad \frac{\partial u}{\partial \mathbf{w}_i} = \frac{1}{\sqrt{W}} (\mathbf{W}_2)_i\, \dot\sigma_{\text{M}}(z_i)\, \boldsymbol{\xi},$$

$$\frac{\partial u}{\partial b_i} = \frac{1}{\sqrt{W}} (\mathbf{W}_2)_i\, \dot\sigma_{\text{M}}(z_i), \qquad \frac{\partial u}{\partial c_k} = \frac{1}{\sqrt{W}} \sum_{i=1}^{W} (\mathbf{W}_2)_i\, \sigma(n_k z_i),$$

$$\frac{\partial u}{\partial n_k} = \frac{1}{\sqrt{W}} \sum_{i=1}^{W} (\mathbf{W}_2)_i\, c_k\, z_i\, \dot\sigma(n_k z_i),$$

where $\dot\sigma_{\text{M}}(z) = \sum_{k=1}^K c_k n_k\, \dot\sigma(n_k z)$ is the derivative of the MFAR activation with respect to its pre-activation. Only the last two gradients involve θ_{freq}; the fixed-frequency network does not possess these parameters, and this is precisely the difference between the two kernels.

Step 2 (kernel assembly). For collocation points $\{\mathbf{x}_m\}_{m=1}^N$ the empirical NTK matrix is $K_\Theta = \sum_\theta \mathbf{g}_\theta \mathbf{g}_\theta^\top$ with $\mathbf{g}_\theta = \big(\partial_\theta u(\mathbf{x}_1), \dots, \partial_\theta u(\mathbf{x}_N)\big)^\top$. Splitting the sum over $\theta = \theta_{\text{base}} \cup \theta_{\text{freq}}$, as in Lemma A.5, gives

$$K_\Theta = K^{\text{fixed}} + \Delta K, \qquad \Delta K = \sum_{k=1}^{K} \Big(\mathbf{g}_{c_k} \mathbf{g}_{c_k}^\top + \mathbf{g}_{n_k} \mathbf{g}_{n_k}^\top \Big), \tag{32}$$

where $K^{\text{fixed}} = \sum_{\theta \in \theta_{\text{base}}} \mathbf{g}_\theta \mathbf{g}_\theta^\top$ is the NTK of the network whose frequency block is frozen at the same parameter values. The block ΔK is therefore the contribution of the trainable frequencies and branch weights alone.

Step 3 (positive semi-definiteness). Every summand of equation 32 is an outer product of a gradient vector with itself, hence positive semi-definite by the argument of Lemma A.5, and a sum of positive semi-definite matrices is positive semi-definite, so $\Delta K \succeq 0$ at any parameter value. The same holds in the infinite-width limit: by the law of large numbers each block converges to its expectation form, in which the frequency contribution is the Gram matrix of the activation Jacobian with respect to θ_{freq}, $\mathbb{E}_{w,b}\big[\mathbf{J}\sigma(z)\mathbf{J}\sigma(z')^\top\big]$, whose columns are the derivatives of the activation with respect to n_k and c_k; a limit of positive semi-definite matrices remains positive semi-definite.

Step 4 (spectral consequence). By Weyl's monotonicity theorem for Hermitian matrices, adding a positive semi-definite matrix cannot decrease any eigenvalue: $\lambda_m(K^{\text{fixed}} + \Delta K) \geq \lambda_m(K^{\text{fixed}})$ for every m in increasing order, and in particular $\lambda_{\min}^+(K^{\text{adapt}}) \geq \lambda_{\min}^+(K^{\text{fixed}})$. In particular the extra block lowers no eigenvalue, and it raises λ_m strictly whenever the corresponding eigenvector $\boldsymbol{\phi}_m$ of K^{fixed} satisfies $\boldsymbol{\phi}_m^\top \Delta K \boldsymbol{\phi}_m > 0$, i.e. whenever it has a nonzero component in the range of ΔK; otherwise the eigenvalue is left unchanged. Under gradient flow, $\frac{d}{dt}\mathbf{u}_t = -K_\Theta(\mathbf{u}_t - \mathbf{y})$; at a fixed parameter value, where the kernel is frozen, the m-th mode decays at rate λ_m, giving $e^{-\lambda_m t}$, and by the inequality above the slowest rate $\lambda_{\min}^+$ is not decreased, so the theoretical worst-case convergence rate under gradient flow is not worsened, and the modes whose eigenvector has a nonzero component in the range of ΔK decay strictly faster. □

B Experiments

B.1 Experimental details

Optimizer: Adam, learning rate 10^{-3}, 150,000 steps, log interval 5,000. Collocation densities and ranks follow the tables: $N_C \in \{64, 128\}$ on the four main benchmarks, with $R \in \{32, 64\}$ for Flow Mixing and $R \in \{16, 32\}$ for two-scale planar transport, Gaussian-vortex transport and 3D Helmholtz; $N_C \in \{32, 64\}$ with $R \in \{32, 64\}$ on Klein–Gordon; and $N_C \in \{12, 24\}$ with $R \in \{6, 12\}$ on the 5D Poisson problem. Test grid 128^3 on the four main benchmarks. Seeds: $445, 987, 123$ (three seeds per configuration). Hardware: NVIDIA RTX 5080 (16GB) via WSL2; JAX/Flax implementation. NTK matrices are computed on the training collocation set; for the architecture comparison we use identical collocation sets and initialization keys and report parameter counts explicitly (353 vs. 1168, and 776 vs. 1168 in the matched-width arm), so the comparison is at comparable, not equal, capacity.

B.2 Additional PDE benchmarks

B.2.1 The four main benchmarks

Flow Mixing. A passive scalar is transported by a rotation whose angular speed decreases with the distance from the origin, so an initially straight interface is wound into a spiral of spatially varying pitch. With $r = \sqrt{x^2 + y^2}$ the problem is

$$\partial_t u + a(r)\, \partial_x u + b(r)\, \partial_y u = 0, \qquad (x, y) \in [-4, 4]^2, \quad t \in [0, 4],$$

driven by the divergence-free rotation field

$$(a, b) = \frac{v_t(r)}{v_{t,\max}} \Big(-\frac{y}{r}, \frac{x}{r} \Big), \qquad v_t(r) = \frac{\tanh r}{\cosh^2 r}, \qquad v_{t,\max} = 0.385,$$

whose angular speed is $\omega(r) = v_t(r)/\big(r\, v_{t,\max}\big)$. Initial values, boundary values and the reference field used for evaluation all come from

$$u(x, y, t) = -\tanh\Big(c\big[y\cos(\omega(r)t) - x\sin(\omega(r)t)\big]\Big),$$

with $c = 0.5$ for Flow Mixing 1 and $c = 1.0$ for Flow Mixing 2, the second setting sharpening the gradients of the interface. The difficulty is the combination of a sharp front with a rotation rate that varies from the core to the exterior: a single carrier frequency matches the local winding rate in one annulus only, and the level sets of the solution are spirals that are aligned with neither coordinate axis.

Two-scale planar transport. On the same domain $(x, y) \in [-4, 4]^2$, $t \in [0, 4]$, a planar wave travelling along the diagonal carries two well-separated spatial scales in a single direction,

$$\partial_t u + \partial_x u + \partial_y u = 0, \qquad u(x, y, t) = \sin g + \tfrac{1}{2}\sin(4g), \qquad g = c\,(x + y - 2t),$$

with $c = 0.5$ for Two-scale planar transport 1 and $c = 1.0$ for Two-scale planar transport 2. The coefficient c scales the two wavenumbers c and $4c$, and with them the two temporal frequencies $2c$ and $8c$, so the target is multi-scale while remaining weakly entangled across the variables: it is exactly separable in the diagonal coordinate g. The benchmark therefore asks whether a factorized representation can carry several harmonics of one direction without inflating its rank.

Gaussian-vortex transport. On the same domain, the same transport operator is driven by a rotation whose rate decays with the radius, so each solution mixes a fast core with a nearly rigid exterior:

$$\partial_t u - \omega(r)\, y\, \partial_x u + \omega(r)\, x\, \partial_y u = 0, \qquad \omega(r) = 2\exp\big(-r^2/4\big),$$
$$u(x, y, t) = -\tanh\big(c\, g\big), \qquad g = \tfrac{1}{2}\big[y\cos(\omega(r)t) - x\sin(\omega(r)t)\big],$$

with $c = 0.5$ for Gaussian-vortex transport 1 and $c = 1.0$ for Gaussian-vortex transport 2. Because the interface is sharp near the origin and almost frozen away from it, the local frequency content spans a much wider range within a single solution than in Flow Mixing, and a model with fixed frequencies has to compromise between the two regimes.

3D Helmholtz. The steady Helmholtz problem separates exactly but oscillates rapidly,

$$\Delta u + k^2 u = q \quad \text{on } [-1, 1]^3, \qquad u\big|_{\partial\Omega} = 0,$$
$$u(x_1, x_2, x_3) = \sin(a_1\pi x_1)\sin(a_2\pi x_2)\sin(a_3\pi x_3),$$

with $(a_1, a_2, a_3) = (2, 2, 1)$ for 3D Helmholtz 1 and $(4, 4, 2)$ for 3D Helmholtz 2, and with the source term $q = \Delta u + k^2 u$ obtained from that solution (the boundary condition holds since $\sin(a_i\pi(\pm 1)) = 0$). Because the problem is steady, the bipartition used here is spatial rather than temporal, $\boldsymbol{\xi} = \{x_1\}$, $\boldsymbol{\eta} = \{x_2, x_3\}$; among the splits of the three coordinates it is the one in which the target is exactly rank-one, so that the low-rank cross-group coupling is not restrictive for it. The benchmark therefore isolates oscillatory content from variable coupling: a model that cannot represent high frequencies fails here even though the target is perfectly separable, and because the grouping confers no structural advantage on this target it also tests how the separation behaves when its prior does not hold.

B.2.2 Solution maps on the main benchmarks

Figures 6 and 7 display the ground-truth solution together with the pointwise absolute error of all twelve models on two of the main benchmarks, Flow Mixing and Gaussian-vortex transport.

B.2.3 Klein–Gordon equation

The nonlinear Klein–Gordon equation is

$$\partial_{tt} u - \Delta u + u^2 - f = 0 \quad \text{on } (x, y) \in [-1, 1]^2, \quad t \in [0, 10], \qquad u(0, x, y) = x + y,$$

and we use the bipartition $\boldsymbol{\xi} = \{t\}$, $\boldsymbol{\eta} = \{x, y\}$. The reference solution is

$$u(t, x, y) = (x + y)\cos(\omega t) + xy\,\sin(\omega t), \qquad f = u^2 - \omega^2 u,$$

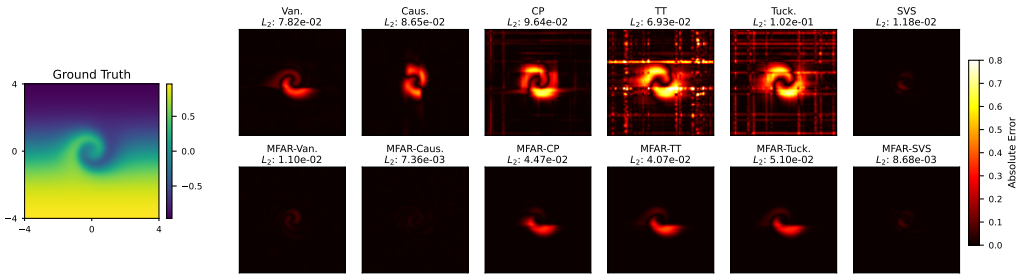


Figure 6: Flow Mixing: ground-truth solution at the evaluation time (left) and the pointwise absolute error of all twelve models (right). In the error block the top row collects the fixed-frequency models and the bottom row their MFAR counterparts, the label above each panel names the model and reports its relative L^2 error, and all error panels share the color scale at the far right. The figure shows *where* each architecture fails, not only by how much: in the fixed-frequency row the error is concentrated in a band that follows the transported interface, and the tensor-decomposed arms additionally show error along the coordinate directions, where the factorization of the individual variables cannot follow the rotation; the MFAR row lowers both the peak amplitude and the spatial extent of that band, and the two SVS panels are almost black, in agreement with their being the two smallest relative L^2 errors of the twelve runs.

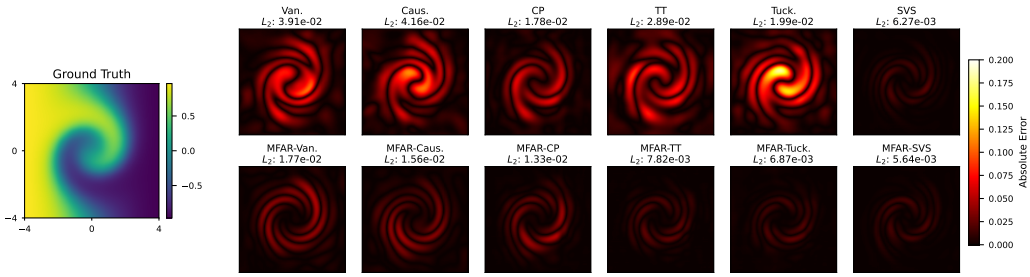


Figure 7: Gaussian-vortex transport: ground-truth solution at the evaluation time (left) and the pointwise absolute error of all twelve models (right), with the same layout as Figure 6. Here the target combines a rapidly rotating core with a nearly rigid exterior, and the error maps reflect that structure: in the fixed-frequency row the error spreads over the whole rotating pattern instead of concentrating on a single interface, with the largest amplitudes near the core; the MFAR row reduces the error along the entire spiral, and the SVS panels are again the two darkest, matching the two smallest relative L^2 errors of the twelve runs. Read together with Figure 6, the two figures indicate that the two mechanisms are complementary: frequency adaptation lowers the error of every model, while the separable arm already starts from a lower error than the other decompositions on both benchmarks.

with $\omega = 2$ for Klein–Gordon 1 and the doubled frequency $\omega = 4$ for Klein–Gordon 2; the source term follows from $u_{xx} = u_{yy} = 0$, which gives $u_{tt} = -\omega^2 u$. Over the bipartition the solution reads $u = \cos(\omega t)\,(x + y) + \sin(\omega t)\,(xy)$, a sum of two separable modes: it is exactly rank-2 in the $\{t\} \times \{x, y\}$ split, and its second spatial mode xy couples x and y multiplicatively, so the target probes intra-group coupling as well. Table 5 summarizes the baseline comparison.

B.2.4 5D Poisson equation

We solve

$$\Delta u = f \quad \text{on } [0,1]^5,$$

$$u = \sum_{i=1}^{5} \sin(\pi x_i/2) \;\; \text{(5D Poisson 1)}, \qquad u = \sum_{i=1}^{5} \sin(\pi x_i) \;\; \text{(5D Poisson 2)},$$

with $f = \Delta u$ obtained from the chosen ground truth. The additive structure is *not* multiplicatively separable, which tests expressivity beyond tensor-product priors, and the bipartite form evaluates

Table 5: Relative L^2 errors on the Klein–Gordon equation across eight configurations, averaged over 3 seeds.

Setting	N_C	R	Vanilla	Causal	CP	Tucker	TT	MFAR-SVS
Klein–Gordon 1	32	32	0.021	0.018	0.032	0.15	0.093	**0.0055**
Klein–Gordon 1	32	64	0.021	0.018	0.055	0.21	0.191	0.0062
Klein–Gordon 1	64	32	0.0097	0.0086	0.172	0.056	0.035	**0.0036**
Klein–Gordon 1	64	64	0.0097	0.0086	0.138	0.065	0.172	0.0047
Klein–Gordon 2	32	32	0.033	0.025	0.052	0.21	0.114	**0.0097**
Klein–Gordon 2	32	64	0.033	0.025	0.075	0.47	0.198	0.014
Klein–Gordon 2	64	32	0.012	0.011	0.028	0.064	0.026	**0.0049**
Klein–Gordon 2	64	64	0.012	0.011	0.032	0.23	0.065	**0.0045**

Table 6: Relative L^2 errors on the 5D Poisson equation across eight configurations, averaged over 3 seeds.

Setting	N_C	R	Vanilla	Causal	CP	Tucker	TT	MFAR-SVS
5D Poisson 1	12	6	0.0069	0.0053	0.0042	0.0039	0.0030	0.00093
5D Poisson 1	12	12	0.0069	0.0053	0.0032	0.0021	0.0020	**0.00092**
5D Poisson 1	24	6	0.00073	0.00087	0.0037	0.0024	0.0031	0.00077
5D Poisson 1	24	12	0.00073	0.00087	0.0028	0.0021	0.0018	**0.00058**
5D Poisson 2	12	6	0.0071	0.0058	0.022	0.020	0.015	**0.0021**
5D Poisson 2	12	12	0.0071	0.0058	0.018	0.017	0.018	0.0022
5D Poisson 2	24	6	0.0025	0.0028	0.019	0.014	0.015	0.0017
5D Poisson 2	24	12	0.0025	0.0028	0.015	0.026	0.013	**0.0017**

the residuals on coordinate grids rather than on the full N_C^5 tensor product. Table 6 reports the errors of both settings.

B.3 Additional results

This appendix collects additional training-dynamics results that complement those of Figure 5.

B.4 Ablation studies

Effect of adaptive frequencies. We first evaluate the contribution of MFAR by replacing the fixed-frequency activations with adaptive frequency representations while keeping the underlying tensor decomposition unchanged. Across the four main benchmarks, learnable frequencies never increase the error of a given decomposition, and they reduce it in 116 of the 156 fixed/adaptive comparisons (Tables 1–4). The reduction is largest for the dimension-wise factorizations on the weakly entangled multi-scale problem, where Tucker improves from 1.15×10^{-1} to 1.7×10^{-2} ($6.8\times$) and CP and TT by up to $1.9\times$ and $1.8\times$; on Flow Mixing the dimension-wise arms gain little and the improvement comes from the structured separation, confirming that the benefit of adaptive frequency learning is not specific to SVS.

Effect of the number of frequency components. We further study the effect of the number of frequency components K in MFAR-SVS (Table 7). Increasing K from 1 to 3 consistently improves the solution accuracy for both fixed and adaptive frequency settings. The adaptive configuration further reduces the best error from 4.11×10^{-3} to 3.87×10^{-3}. Increasing K beyond 3 provides diminishing accuracy gains while increasing the training cost. Therefore, we use $K = 3$ as the default setting throughout the experiments.

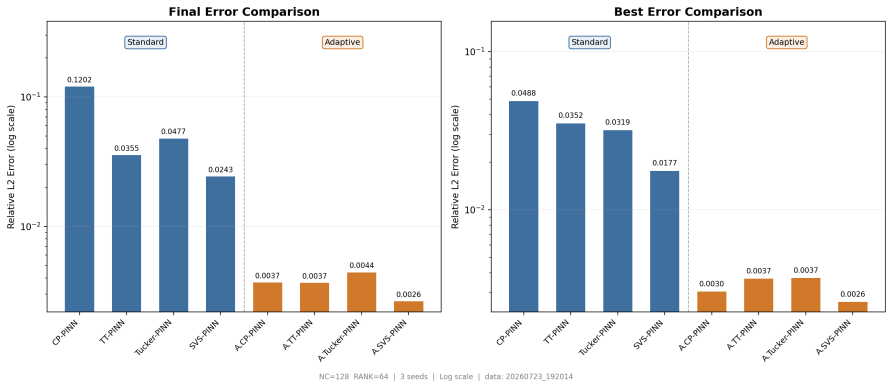


Figure 8: Fixed versus adaptive multi-frequency activations across four tensor decompositions, reported separately for the final error of each run (left) and the best error reached during training (right). Within each panel the first four bars are the fixed-frequency factorizations (CP, TT, Tucker) together with the separable SVS model, and the last four bars are the same models with adaptive multi-frequency activations; the vertical axis is the relative L^2 error on a logarithmic scale. Two observations follow. Making the frequencies learnable lowers the error for every decomposition and in both panels, so the benefit is not specific to the SVS structure; and the SVS model is the most accurate member of the fixed-frequency group in both panels, while its adaptive counterpart is the most accurate overall, indicating that variable separation and frequency adaptation compose rather than substitute for one another.

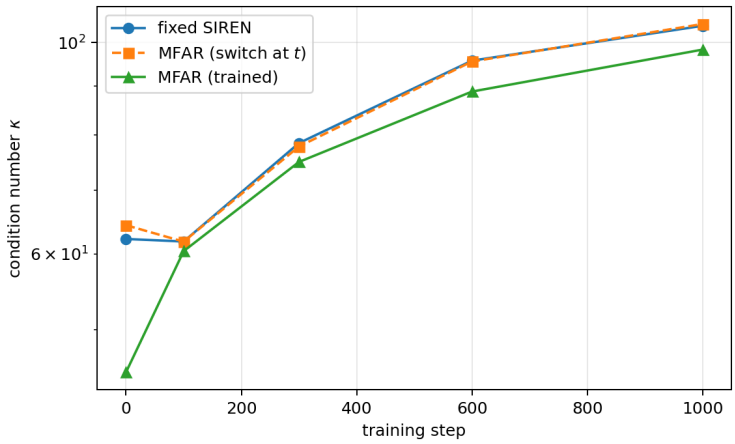


Figure 9: Evolution of the NTK condition number κ during training for three frequency treatments of the same architecture: fixed sinusoidal activations, a variant in which the adaptive frequencies are switched on only from an intermediate step onward, and fully trained adaptive frequencies. The vertical axis is logarithmic. All three curves rise as training proceeds, that is, the kernel spectrum becomes progressively more imbalanced; but the fully trained variant stays at or below the other two throughout and clearly beneath them once training is under way, whereas the variant that switches the frequencies on later tracks the fixed-frequency baseline almost exactly. The reduction is therefore attributable to the frequency parameters being optimized throughout training, and not merely to the presence of a multi-frequency activation.

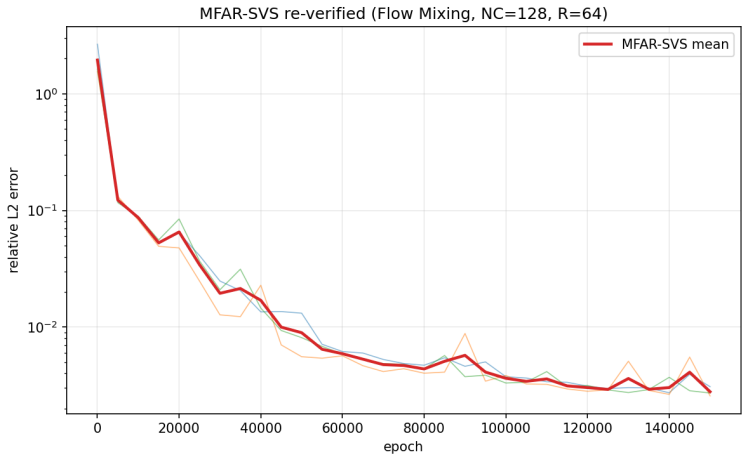


Figure 10: Convergence of MFAR-SVS on Flow Mixing over three independent seeds: the thick curve is the mean relative L^2 error and the thin curves are the individual runs, on a logarithmic vertical axis. The error falls by about two orders of magnitude early in training and then decreases slowly and monotonically to its final level, with no plateau that would indicate stalled optimization; the three runs stay close to one another throughout, so the accuracy reported in the tables is not a single favourable draw.

Table 7: Ablation on the number of frequency components for Flow Mixing. Best rel. L^2 errors and training time are reported; bold marks the default $K = 3$ setting used throughout.

	Fixed		Trainable			
K	1	3	1	3	5	7
Best rel. L^2	1.45e−2	4.11e−3	1.82e−2	**3.87e−3**	3.94e−3	3.36e−3
Final	4.87e−2	4.11e−3	2.93e−2	3.87e−3	2.81e−2	4.06e−3
Time (s)	196.4	472.6	225.6	512.6	926.6	1254.5