CAUCHY ACTIVATIONS FOR DIRECT SOLUTION OF LINEAR PDES

Anonymous authors
Paper under double-blind review

ABSTRACT

Cauchy activations provide an analytic trial space for solving linear partial differential equations by direct least squares. We develop CAUCHY FEATURE COLLOCATION (CFC), which fixes complex poles outside the physical domain and recovers output coefficients from PDE, boundary, and initial constraints. Closed-form mixed derivatives assemble the operator matrices, and a singular-value decomposition determines the coefficients without gradient descent. For targets holomorphic on a neighborhood containing the pole contours, we prove geometric approximation of a fixed finite jet with shared coefficients. Output-specific identities describe recovery from the fitted constraints. Controlled heat and plate experiments show that direct Cauchy fitting achieves lower field and differential errors than coefficient-only Adam under identical matrices and objectives at the tested optimization budget. Comparisons with fixed tanh, sine, Gaussian RBF, and Chebyshev dictionaries using a common direct solver reveal a problem-dependent ordering: Cauchy attains the lowest heat differential error, while sine features lead on plate accuracy. In a separate study of sparse values and slopes across analytic near-singularity families, Cauchy-Complex obtains the lowest aggregate error for held-out higher derivatives.

1 INTRODUCTION

Solving a differential equation with a neural representation couples two choices: the functions used to represent the solution and the procedure used to fit their coefficients. Physics-informed neural networks (PINNs) learn a field by minimizing equation and boundary residuals (Raissi et al., 2019; Karniadakis et al., 2021). For a linear PDE, fixing the features makes those residuals linear in the output coefficients. The resulting quadratic objective admits a direct least-squares solve. This structure lets the representation carry the approximation task while numerical linear algebra determines the coefficients.

We develop CAUCHY FEATURE COLLOCATION (CFC) around Cauchy activations. Each feature is a product of reciprocal factors $(\xi_j - x_j)^{-1}$, with complex poles separated from the physical domain. Pole locations define the trial space and remain fixed during coefficient fitting. Differentiation raises the powers of these factors and multiplies them by known factorials. PDE, boundary, and initial constraints therefore become explicit matrix rows. One singular-value decomposition yields the regularized or truncated least-squares solution, and the same coefficient vector evaluates temperature, displacement, flux, curvature, or load.

This construction has concrete computational benefits for linear PDEs. The coefficient objective is convex, with a unique minimizer under positive ridge regularization and a specified minimum-norm solution under singular-value truncation. Its solution is independent of a learning-rate schedule or an initialization of trainable weights. Spatial and temporal derivatives are assembled directly from the activation formula, including the fourth-order and mixed derivatives needed for plate equations. The resulting field remains differentiable throughout the domain, with each additional output evaluated by a matrix-vector product.

Direct output-weight fitting also underlies extreme learning machines, physics-informed extreme learning machines, and random-feature PDE methods (Huang et al., 2006; Dwivedi & Srinivasan, 2020; Chen et al., 2022). CFC develops this approach with a deterministic exterior-pole dictionary and Cauchy differentiation. The poles expose the continuation geometry of the approximation, connecting the representation to rational collocation and exterior-pole solvers (Baltensperger & Berrut, 2001;

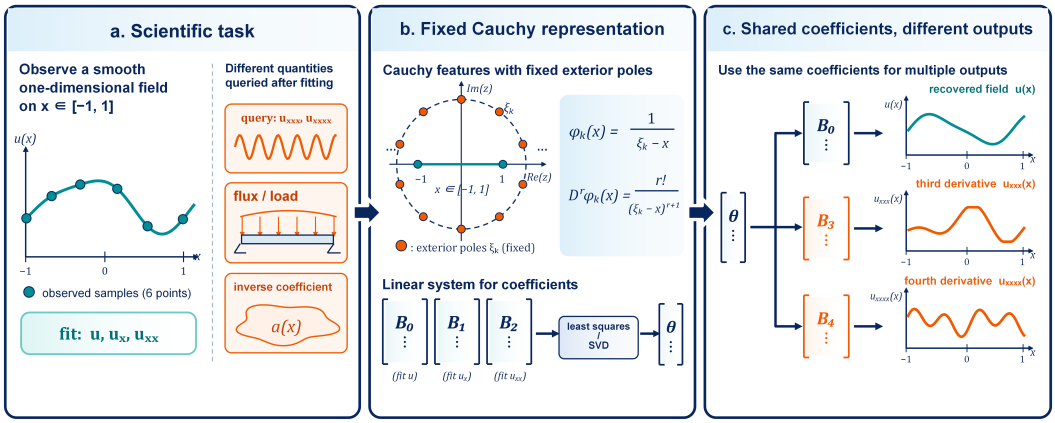


Figure 1: **A shared Cauchy representation for differential outputs.** The original schematic illustrates fitting values, slopes, and curvature, then applying further derivative matrices to the same coefficient vector. The PDE studies fit equation and boundary constraints; the sparse-data study fits values and slopes. Flux, load, and coefficient recovery in the schematic illustrate possible applications of differential outputs.

Costa & Trefethen, 2023; Gopal & Trefethen, 2019). Our analysis makes this connection precise: a Cauchy quadrature theorem constructs a common approximant for a prescribed finite collection of derivatives, and recovery identities describe how the fitted constraints determine each requested output. A characterization of the observation nullspace explains the remaining coefficient choices under sparse data.

The experiments address the solver and the representation separately. On heat and biharmonic plate equations, a controlled study holds the Cauchy matrix and objective fixed and compares direct fitting with coefficient-only Adam at a fixed optimization budget. A further control fits fixed tanh, sine, Gaussian RBF, and Chebyshev dictionaries with the same direct solver, observation budget, and real column count. Cauchy gives lower field and differential errors than tanh and RBF on both problems; it leads heat differential accuracy, while sine leads plate accuracy. An end-to-end PINN comparison provides a further reference under its stated training budget. Finally, competing representations receive identical sparse values and slopes. On new instances of analytic near-singularity families, Cauchy-Complex has the lowest aggregate error at every queried derivative order. Together, these studies identify the contribution of direct coefficient recovery and the target-dependent accuracy of the rational trial space.

2 Related Work

Direct coefficient fitting for PDEs. Extreme learning machines fit output weights in a fixed feature map (Huang et al., 2006). Physics-informed extreme learning machines apply this principle to stationary and time-dependent linear PDEs (Dwivedi & Srinivasan, 2020). Random-feature PDE methods combine fixed features, collocation, residual scaling, and multiscale representations (Chen et al., 2022). These methods establish direct least squares as a practical fitting rule. CFC uses a deterministic rational dictionary whose pole geometry and explicit mixed derivatives connect this rule to Cauchy approximation. Cauchy activation networks learn rational units (Li et al., 2025), while CauchyNet learns complex shifts and output weights in the reciprocal-product kernel family (Zhang et al., 2026). Here the shifts remain fixed during the coefficient solve. Cauchy random features sample Fourier frequencies for Sobolev operator learning (Liao et al., 2025); CFC uses rational activations and fits each equation instance directly.

Cauchy and rational approximation. Cauchy differentiation and periodic quadrature achieve geometric convergence inside an analytic domain (Lyness & Moler, 1967; Bornemann, 2011; Austin et al., 2014; Trefethen & Weideman, 2014). They construct approximants from contour values. Here the contour defines a trial space whose coefficients are recovered from real-domain constraints. Ratio-

nal collocation, AAA–least squares, and lightning solvers provide related structured discretizations (Baltensperger & Berrut, 2001; Costa & Trefethen, 2023; Gopal & Trefethen, 2019). AAA adaptively selects a barycentric rational approximant (Nakatsukasa et al., 2018); our function comparison includes an adapter that uses its selected poles to refit both value and slope observations. Our analysis follows a common coefficient vector across a finite collection of derivatives and describes its recovery under ridge or singular-value truncation.

Learning from differential information. PINNs fit neural fields through PDE, boundary, and observation residuals (Raissi et al., 2019; Karniadakis et al., 2021). Sobolev training and gradient-enhanced PINNs add derivative observations to the objective (Czarnecki et al., 2017; Yu et al., 2022), and loss balancing addresses optimization across residual blocks (Wang et al., 2021). Neural operators learn maps across problem instances (Lu et al., 2021; Li et al., 2021); derivative-informed neural operators also learn parametric Jacobians (O'Leary-Roseberry et al., 2024). Kernel interpolation from linear functionals (Schaback & Wendland, 2006) and goal-oriented approximation (Oden & Prudhomme, 2001; Brevis et al., 2024) motivate assessing the fitted solution at its requested output. We report both fields and differential quantities, with observations held fixed across the compared methods.

3 A DIRECT SOLVER WITH CAUCHY ACTIVATIONS

Linear equations and constraints. Let $u : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ satisfy $\mathcal{L}u = f$ in Ω, together with linear boundary or initial constraints $\mathcal{C}_\ell u = g_\ell$. The operator $\mathcal{L} = \sum_\alpha q_\alpha(\boldsymbol{x}) D^\alpha$ has known coefficients. CFC chooses fixed features, assembles these constraints at collocation sites, and fits their output coefficients. Pole geometry and regularization are selected on development data and fixed for evaluation; the residual-block weights are prescribed by the protocol.

Cauchy activations and analytic derivatives. Choose pole vectors $\boldsymbol{\xi}_k \in \mathbb{C}^d$ satisfying $\inf_{x_j \in \pi_j(\Omega)} |\xi_{k,j} - x_j| > 0$ for every k, j. Define

$$\phi_k(\boldsymbol{x}) = \prod_{j=1}^{d} (\xi_{k,j} - x_j)^{-1}, \qquad \widehat{u}(\boldsymbol{x}) = a_0 + \operatorname{Re} \sum_{k=1}^{m} c_k \phi_k(\boldsymbol{x}), \tag{1}$$

where $a_0 \in \mathbb{R}$ and $c_k \in \mathbb{C}$. In one dimension, a pole $\xi = a + \mathrm{i}b$ gives the two real activations $(a - x)/((a - x)^2 + b^2)$ and $b/((a - x)^2 + b^2)$, the real and negative imaginary parts of $(\xi - x)^{-1}$. Products extend the construction to several coordinates. Every mixed derivative has the explicit form

$$D^\alpha \phi_k(\boldsymbol{x}) = \alpha! \prod_{j=1}^{d} (\xi_{k,j} - x_j)^{-(\alpha_j + 1)}, \qquad \alpha! = \prod_{j=1}^{d} \alpha_j!. \tag{2}$$

Thus derivative assembly uses reciprocal powers and known constants. If Φ_α stores these complex values, let $\boldsymbol{b}_\alpha = \mathbf{1}$ for $\alpha = \mathbf{0}$ and $\boldsymbol{b}_\alpha = \mathbf{0}$ otherwise. Realification gives

$$B_\alpha = [\boldsymbol{b}_\alpha \ \operatorname{Re} \Phi_\alpha \ -\operatorname{Im} \Phi_\alpha], \qquad D^\alpha \widehat{\boldsymbol{u}} = B_\alpha \boldsymbol{\theta}, \quad \boldsymbol{\theta} = [a_0; \operatorname{Re} \boldsymbol{c}; \operatorname{Im} \boldsymbol{c}]. \tag{3}$$

The real-part-only variant keeps the real columns; setting $a_0 = 0$ removes the constant. The PDE study uses both of these restrictions.

Assembling the equation. At interior sites $\boldsymbol{x}_n$, form

$$A_{\mathcal{L}} = \sum_\alpha \operatorname{diag}(q_\alpha(\boldsymbol{x}_n)) B_\alpha. \tag{4}$$

The heat operator uses $B_{(1,0)} - \kappa B_{(0,2)}$, and the plate operator uses $B_{(4,0)} + 2B_{(2,2)} + B_{(0,4)}$. Boundary and initial operators give further blocks in the same way. For blocks $(A_b, \boldsymbol{y}_b)$ with n_b rows and weights $w_b > 0$, stack

$$A = \begin{bmatrix} \sqrt{w_1/n_1} A_1 \\ \vdots \\ \sqrt{w_K/n_K} A_K \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \sqrt{w_1/n_1} \boldsymbol{y}_1 \\ \vdots \\ \sqrt{w_K/n_K} \boldsymbol{y}_K \end{bmatrix}. \tag{5}$$

Any column scaling is carried through to every queried block, so fitted and queried matrices use the same coefficient coordinates.

One coefficient solve. For positive ridge, the output head is the unique global minimizer

$$\widehat{\boldsymbol{\theta}}_\lambda = \arg\min_{\boldsymbol{\theta}} \|A\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2. \tag{6}$$

Using a thin SVD $A = U\,\mathrm{diag}(\sigma_j)V^\top$ and a cutoff $\tau \geq 0$, compute

$$\widehat{\boldsymbol{\theta}} = V\,\mathrm{diag}(g_j)U^\top \boldsymbol{y}, \qquad g_j = \begin{cases} \sigma_j/(\sigma_j^2 + \lambda), & \lambda > 0, \\ 1/\sigma_j, & \lambda = 0,\ \sigma_j > \tau, \\ 0, & \lambda = 0,\ \sigma_j \leq \tau. \end{cases} \tag{7}$$

At zero ridge, this is the minimum-norm least-squares solution restricted to the retained right-singular subspace. Each linear PDE fit consists of matrix assembly and this factorization, followed by direct evaluation $B_\alpha\widehat{\boldsymbol{\theta}}$. For a fixed operator, sites, geometry, scales, and spectral filter, the factorization also serves multiple forcing or boundary-data vectors through new products $U^\top\boldsymbol{y}$. This reuse is an algebraic property of the solver.

Geometry, capacity, and cost. Circular dictionaries use centers c_j, radii ρ_j, node counts m_j, and phases $\vartheta_j \in [0, 1)$:

$$\xi_{k_j,j} = c_j + \rho_j e^{2\pi\mathrm{i}(k_j+\vartheta_j)/m_j}, \qquad k_j = 0, \ldots, m_j - 1. \tag{8}$$

Within the analytic regime of Theorem 4.1, radius balances separation from the real domain against distance to the analyticity boundary. The function study's selected Cauchy-Complex model uses 12 upper-half-plane poles and a constant: 25 real coefficients. Cauchy-Real uses 64 real-part full-circle columns and a constant, spanning 33 independent functions because conjugate real-part columns coincide. The PDE models use real-part tensor features with $a_0 = 0$. The PINN comparison reports stored columns and independent capacities in Section 5.1; the controls in Section 5.2 compress conjugate pairs and normalize every dictionary before fitting.

For n fitted rows and p real columns, dense assembly stores $O(np)$ numbers and SVD costs $O(np\min\{n, p\})$. Once factored, each new right-hand side costs $O((n + p)\min\{n, p\})$; evaluation at q sites costs $O(qp)$ after constructing the queried matrix. A tensor dictionary has $\prod_j m_j$ complex features, making this implementation suited to low-dimensional problems. The experiments use one and two coordinates.

Field and differential accuracy. For an evaluation grid X and a nonzero reference norm, define

$$E_\alpha = \frac{\|D^\alpha\widehat{u}(X) - D^\alpha u(X)\|_2}{\|D^\alpha u(X)\|_2}. \tag{9}$$

The PDE study reports field and governing differential errors. The representation study fits orders $\{0, 1\}$ and queries orders $\{2, 3, 4\}$ from the same fitted model, testing derivative prediction under a common sparse observation budget.

4 Approximation and Recovery for the Direct Solver

The trial space and the coefficient solve contribute separately to accuracy. We first establish simultaneous approximation of the field and its derivatives, then describe recovery from the assembled constraints. Complete proofs appear in Appendix A.

A common reference approximant. The derivative identity (2) preserves the coefficient coordinates. Cauchy quadrature supplies one vector approximating every derivative in a prescribed finite jet.

Theorem 4.1 (Shared-coefficient Cauchy approximation for a finite jet)**.** *Let $\boldsymbol{c} \in \mathbb{R}^d$, let $M = \prod_{j=1}^d \{x_j \in \mathbb{R} : |x_j - c_j| \leq r_j\}$, and let f be holomorphic on a neighborhood of $\{\boldsymbol{z} \in \mathbb{C}^d : |z_j - c_j| \leq R_j,\ j = 1, \ldots, d\}$. Suppose $0 < r_j < s_j < \rho_j < R_j$. For positive integers m_j, write $\boldsymbol{m} = (m_1, \ldots, m_d)$ and $\mathcal{K}_{\boldsymbol{m}} = \prod_{j=1}^d \{0, \ldots, m_j - 1\}$. For fixed phases $\vartheta_j \in [0, 1)$ and $\boldsymbol{k} \in \mathcal{K}_{\boldsymbol{m}}$, set*

$$t_{k_j,j} = \frac{2\pi(k_j + \vartheta_j)}{m_j}, \qquad \xi_{k_j,j} = c_j + \rho_j e^{\mathrm{i}t_{k_j,j}}, \tag{10}$$

define $\boldsymbol{\xi_k} = (\xi_{k_1,1}, \ldots, \xi_{k_d,d})$*, and set*

$$\gamma_{\boldsymbol{k}} = f(\boldsymbol{\xi_k}) \prod_{j=1}^{d} \frac{\rho_j e^{\mathrm{i}t_{k_j,j}}}{m_j}. \tag{11}$$

For every fixed $a \in \mathbb{N}_0$ *and every* $\alpha \in \mathbb{N}_0^d$ *with* $|\alpha| \le a$*,*

$$\sup_{\boldsymbol{x}\in M} \left| D^\alpha f(\boldsymbol{x}) - \alpha! \sum_{\boldsymbol{k}\in\mathcal{K}_{\boldsymbol{m}}} \gamma_{\boldsymbol{k}} \prod_{j=1}^{d} (\xi_{k_j,j} - x_j)^{-(\alpha_j+1)} \right| \le C_\alpha \sum_{j=1}^{d} \left[\left(\frac{s_j}{\rho_j}\right)^{m_j} + \left(\frac{\rho_j}{R_j}\right)^{m_j} \right], \tag{12}$$

where C_α *is independent of* $\boldsymbol{m}$*.*

The first geometric term measures separation of the real box from the differentiated kernels; the second measures separation of the contour from the analyticity boundary. The constants depend on the target, geometry, dimension, and derivative order. For each fixed finite jet, both geometric terms decrease with the number of contour nodes.

The coefficients $\gamma_{\boldsymbol{k}}$ construct a reference approximant from complex-contour values. The implemented head recovers coefficients from real observations. For targets real-valued on M, taking the real part preserves the bound; an equivalent compact realification of a conjugate-paired rule does so as well. Restricting the coefficient components changes the approximation space. The geometric rate applies when the contour lies inside the target's holomorphic neighborhood, as required in the theorem.

Recovering a queried output. For any fixed scaled design, the next identity separates the effects of fitted-data error, coefficient recovery, and output approximation. It applies to both implemented linear heads and to every selected geometry.

Proposition 4.2 (Output error for ridge and truncated SVD)**.** *Let* $A \in \mathbb{R}^{n\times p}$*,* $B \in \mathbb{R}^{q\times p}$*,* $\boldsymbol{z} \in \mathbb{R}^q$*, and* $\boldsymbol{\theta}_m \in \mathbb{R}^p$*. Write* $\boldsymbol{y} = A\boldsymbol{\theta}_m + \boldsymbol{e}$ *and* $\boldsymbol{r}_B = B\boldsymbol{\theta}_m - \boldsymbol{z}$*. For* $\lambda > 0$*, let* $H_\lambda = A^\top A + \lambda I$ *and* $\widehat{\boldsymbol{\theta}}_\lambda = H_\lambda^{-1} A^\top \boldsymbol{y}$*. Then*

$$B\widehat{\boldsymbol{\theta}}_\lambda - \boldsymbol{z} = BH_\lambda^{-1}A^\top \boldsymbol{e} - \lambda BH_\lambda^{-1}\boldsymbol{\theta}_m + \boldsymbol{r}_B, \tag{13}$$

and

$$\|B\widehat{\boldsymbol{\theta}}_\lambda - \boldsymbol{z}\|_2 \le \|BH_\lambda^{-1}A^\top\|_2\|\boldsymbol{e}\|_2 + \lambda\|BH_\lambda^{-1}\|_2\|\boldsymbol{\theta}_m\|_2 + \|\boldsymbol{r}_B\|_2. \tag{14}$$

For $\tau \ge 0$*, let* $A_\tau^\dagger$ *be the pseudoinverse formed from singular directions of* A *whose singular values exceed* τ*. Define* $P_\tau = A_\tau^\dagger A$ *and set* $\widehat{\boldsymbol{\theta}}_\tau = A_\tau^\dagger \boldsymbol{y}$*. Then*

$$B\widehat{\boldsymbol{\theta}}_\tau - \boldsymbol{z} = BA_\tau^\dagger \boldsymbol{e} + B(P_\tau - I)\boldsymbol{\theta}_m + \boldsymbol{r}_B, \tag{15}$$

and

$$\|B\widehat{\boldsymbol{\theta}}_\tau - \boldsymbol{z}\|_2 \le \|BA_\tau^\dagger\|_2\|\boldsymbol{e}\|_2 + \|B(P_\tau - I)\boldsymbol{\theta}_m\|_2 + \|\boldsymbol{r}_B\|_2. \tag{16}$$

Here $\boldsymbol{e}$ contains fitted-row approximation and observation error, $\boldsymbol{r}_B$ is the target-output approximation residual, and the middle terms measure ridge or truncation bias. Choosing $B = B_{\mathbf{0}}$ gives field recovery; choosing $B = B_\alpha$ gives a derivative output. Linear combinations of these matrices give the heat and plate operator outputs in Equation (4). For the zero-ridge head, the query-specific perturbation factor is $\|B_\alpha A_\tau^\dagger\|_2$. The projection term accounts for unresolved coefficient directions, and $\boldsymbol{r}_B$ accounts for approximation of the requested output. Relative error divides all terms by the reference output norm, which itself changes with derivative order.

The identities give a direct interpretation of representation choice: pole geometry determines the approximation space and the matrices A and B, while ridge or truncation determines the recovery map. Prediction at a new derivative order depends on all three terms.

Finite observations and ambiguity. For distinct sites $x_1, \ldots, x_n$ and $s \in \mathbb{N}_0$, let $q_s(x) = \prod_{i=1}^n (x - x_i)^{s+1}$. Then $q_s^{(j)}(x_i) = 0$ for $0 \le j \le s$, whereas

$$q_s^{(s+1)}(x_i) = (s+1)! \prod_{j\ne i} (x_i - x_j)^{s+1} \ne 0. \tag{17}$$

Problem	Method	Field error	Differential error	Training loss
Heat	CFC	3.38×10^{-8}	1.76×10^{-6}	6.70×10^{-15}
Heat	tanh PINN	7.53×10^{-3}	1.37×10^{-1}	7.16×10^{-4}
Plate	CFC	5.63×10^{-8}	4.13×10^{-6}	1.92×10^{-11}
Plate	tanh PINN	8.24×10^{-3}	1.02×10^{-2}	2.92×10^{-3}

Table 1: **Five-seed median PDE errors.** Both methods use common training points, block weights, and evaluation grids. Differential errors concern u_{xx} for heat and $\Delta^2 u$ for plate. Training loss is the weighted mean residual after fitting; the CFC ridge penalty is reported through its configuration. PINN values follow 2,000 Adam updates.

Thus f and $f + \eta q_s$ agree on the observed jets through order s and differ at the next order. The sparse-observation protocol uses $s = 1$.

This ambiguity also occurs inside the main Cauchy space. For m distinct upper-half-plane poles, its constant and $2m$ realified features span

$$\mathcal{V}_m = \{P/Q_\xi : \ P \in \mathbb{R}[x], \ \deg P \le 2m\}, \qquad Q_\xi(x) = \prod_{k=1}^{m} (x - \xi_k)(x - \overline{\xi_k}).$$

When $n \le m$, the nonzero member $h_n(x) = \prod_{i=1}^{n} (x - x_i)^2 / Q_\xi(x)$ has zero observed values and first derivatives and strictly positive second derivatives at all sites. The observation map has nullity $\max\{2m + 1 - 2n, 0\}$ (Proposition A.1). The selected 25-coefficient head at $N = 12$ therefore has a one-dimensional observation nullspace. The regularized or minimum-norm head specifies how this remaining direction is resolved.

5 Experiments

We evaluate direct PDE solution, isolate the effects of coefficient fitting and feature choice, and test higher-derivative prediction from sparse observations. All studies use float64. The numerical tables report development-selected comparisons; the original figures retain their fixed configurations, specified in Appendix C. The PINN and function comparisons use evaluation seeds $\{101, 211, 307, 401, 503\}$; the solver controls use new seeds specified below. Exact targets and configurations appear in Appendices B.1, D.1, D.2 and E.

5.1 Direct solution of heat and plate equations

The matched-loss study fits $\sum_{b=1}^{3} w_b n_b^{-1} \|r_b\|_2^2$ with weights $(1, 50, 50)$, shared training points, and common evaluation grids. CFC determines its coefficients through one SVD; a 2–64–64–64–1 tanh PINN updates all 8,577 parameters for 2,000 full-batch Adam steps. Development selection compares 27 CFC geometry/ridge settings and three PINN learning rates on two sampling seeds, using independent equation and boundary/initial residuals. The selected settings remain fixed for five evaluation seeds; Appendix D.2 specifies every configuration.

Heat equation. We solve $u_t = 0.2 u_{xx}$ on $[0, 2] \times [0, \pi]$, with $u(0, x) = \sin x$, zero spatial boundary data, and exact solution $u^\star = \sin x \, e^{-0.2t}$. Evaluation reports errors in u and u_{xx}.

Biharmonic plate. We solve $\Delta^2 u = f$ on $[0, \pi]^2$ with $u = \Delta u = 0$ on the boundary. The exact solution $u^\star = \sin x \sin y + 0.05 \sin(3x) \sin(2y)$ gives $f = 4 \sin x \sin y + 8.45 \sin(3x) \sin(2y)$. Evaluation reports errors in u and $\Delta^2 u$. The selected CFC grids have 18×18 poles for heat and 22×22 for plate, with real spans of at most 162 and 242. Both PINNs select learning rate 0.001.

Accuracy after direct coefficient recovery. CFC achieves lower field and differential errors on both smooth problems at these solver budgets (Table 1). The field errors measure solution recovery on the evaluation grid; differential errors measure the operator's derivative quantities. The following controls separate coefficient fitting from feature choice.

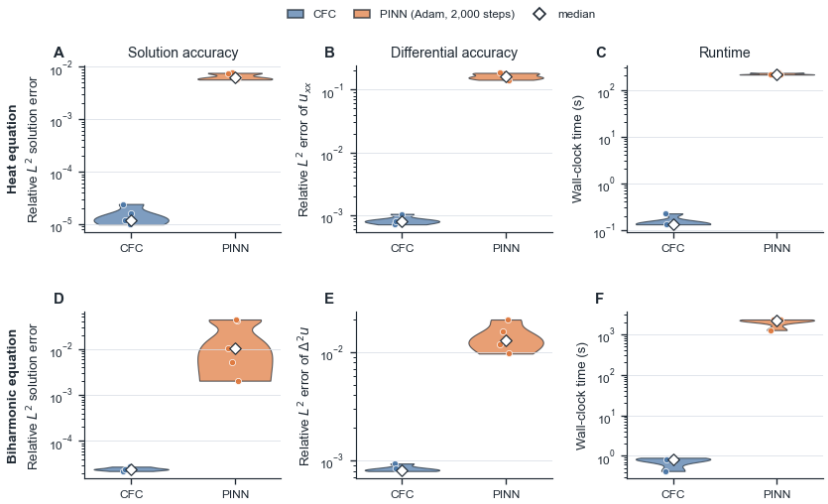


Figure 2: **Original fixed-configuration heat and plate comparison.** Each violin contains five runs; dots show individual runs and diamonds mark medians. Differential errors concern u_{xx} and $\Delta^2 u$. This supplied figure uses the original $14 \times 14/18 \times 18$ CFC grids, ridge 10^{-10}, and a 2,000-update PINN at rate 0.002. Original loss conventions, method-specific evaluation grids, and total runtimes are specified in Appendix C; Table 1 reports the additional matched-loss study.

	Heat			Plate		
Method	Field	u_{xx}	Fit (ms)	Field	$\Delta^2 u$	Fit (ms)
Cauchy / SVD	7.64×10^{-12}	$\mathbf{1.50\times10^{-10}}$	15.1	5.17×10^{-8}	4.07×10^{-6}	27.9
Cauchy / Adam	8.63×10^{-3}	7.37×10^{-2}	541.2	6.65×10^{-2}	4.01×10^{-1}	636.7
Fixed tanh / SVD	1.06×10^{-9}	1.11×10^{-7}	27.0	4.47×10^{-7}	1.04×10^{-5}	49.3
Fixed sine / SVD	4.66×10^{-12}	3.31×10^{-10}	13.5	$\mathbf{2.61\times10^{-8}}$	$\mathbf{8.03\times10^{-7}}$	25.6
Gaussian RBF / SVD	1.00×10^{-6}	2.97×10^{-5}	33.9	4.34×10^{-5}	7.72×10^{-4}	63.4
Chebyshev / SVD	$\mathbf{1.68\times10^{-12}}$	3.36×10^{-10}	14.6	2.32×10^{-6}	7.20×10^{-5}	25.1

Table 2: **Controlled feature and solver comparisons.** Five-seed median relative errors and serial fit times, including matrix assembly. Every method uses 162 normalized columns for heat and 242 for plate. Cauchy/Adam receives the same matrix, right-hand side, and ridge as Cauchy/SVD and takes 20,000 updates. Bold marks the lowest error among the direct methods. These newly selected, normalized configurations define a separate protocol from Table 1.

Figure 2 retains the original fixed-configuration study. Its reported errors and runtimes belong to that protocol. The matched-loss results in Table 1 use the development-selected configurations and common grids described above.

5.2 Separating coefficient fitting from representation

We run a controlled study on the same PDEs with five new sampling seeds $\{809, 811, 821, 823, 827\}$. Every dictionary has 162 real columns for heat and 242 for plate. Cauchy retains one representative of each conjugate pair of its real-part tensor features. All methods use the same training points, residual weights, column normalization, and SVD implementation. The alternatives are fixed tanh, fixed sine, Gaussian RBF, and tensor Chebyshev features. Development selection compares 27 scale/ridge configurations per activation dictionary and three ridge values for the fixed Chebyshev basis, using independent equation and boundary residuals.

The coefficient solver. After freezing the Cauchy dictionary, we fit its coefficients with full-batch Adam from zero initialization. Its learning rate is selected on development residuals and held constant throughout each evaluation run. Table 2 shows that direct fitting achieves substantially lower field

Method	Right	Two-sided	Log	Tanh	Overall GM
Cauchy-Complex	0.828	**0.801**	**0.700**	**0.370**	**0.623**
Cauchy-Real	0.959	1.160	1.788	1.312	1.228
Chebyshev	1.000	1.000	1.001	1.000	1.000
Fourier	1.000	1.000	1.009	1.001	1.003
Gaussian RBF	0.986	0.999	0.947	0.966	0.974
AAA-pole refit	**0.224**	159.378	1.116	1.000	1.286
tanh MLP	0.918	1.427	0.961	0.977	1.043
CFC (original configuration)	0.676	1.023	0.894	0.904	0.804

Table 3: **Fourth-derivative prediction from 12 value/slope sites.** Family columns average E_4 over three new instances and five sampling seeds. Overall GM averages seeds within each instance, then takes the geometric mean over all 12 instances. Bold marks the lowest point estimate among the seven selected methods. The last row evaluates the original CFC configuration on the same instances.

and differential errors in less fitting time under the stated budgets. Every recorded training state is retained in the artifact; Appendix E specifies the optimization settings. This comparison isolates the fitting rule for the selected Cauchy matrices, with the SVD's declared singular-value truncation.

The feature dictionary. With the coefficient solver held fixed, Cauchy gives smaller field and operator errors than tanh and Gaussian RBF on both problems. It has the lowest heat differential error; Chebyshev has the lowest heat field error, and fixed sine has the lowest plate errors. These outcomes identify a strong benefit from direct fitting and a problem-dependent ordering of the trial spaces. Appendix E gives every selected setting, seedwise accuracy, and the numerical diagnostics.

5.3 Predicting higher derivatives from sparse observations

The representation study uses four analytic families on $[-1, 1]$: right-endpoint multipoles, two-sided multipoles, logarithmic branch functions, and tanh endpoint layers. Eight development functions and 12 new evaluation functions contain two and three instances per family, respectively. Every method fits the same values and slopes at $N = 12$ stratified sites and is queried at orders $2, 3, 4$ on a common 2,048-point grid. Each method has 32 candidate configurations, selected by the geometric mean of development E_4 after averaging two sampling seeds. Evaluation uses five seeds and one frozen configuration per method.

The selected Cauchy-Complex model has 12 upper-half-plane poles, radius 1.05, 25 real coefficients, and zero ridge. Exact targets, scaling, capacities, and configurations for all seven methods appear in Appendices B.1 and D.1.

Cauchy-Complex has the lowest aggregate error at each queried order (Table 6), with Gaussian RBF the strongest alternative for the highest-order output. It leads on the two-sided multipole, logarithmic, and tanh family means, while AAA-pole refitting leads on right multipoles. Development selection lowers CFC's aggregate error relative to its original configuration, while paired instance results show both gains and regressions. The improvement therefore depends on the target. Appendix D.1 reports all queried orders and the reference-function comparisons; the artifact retains every paired instance result and the training and observation-budget records.

Original reference-function figures. Figures 3 and 4 retain the original two-sided multipole and tanh comparisons. Their fixed configurations and reference functions differ from the development-selected, new-instance comparison in Table 3. Appendix C specifies the original fitting and timing conventions.

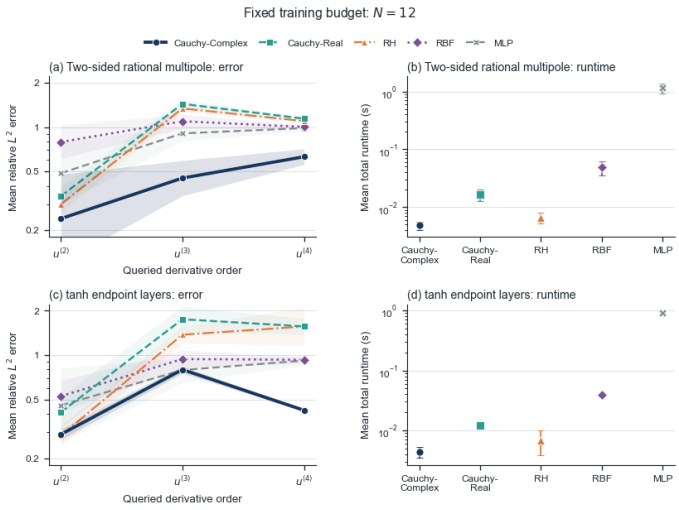


Figure 3: **Original derivative comparison at $N = 12$.** The two-sided multipole and tanh targets are fitted using values and slopes. Left: five-seed mean relative errors at orders two, three, and four, with multiplicative log-standard-deviation bands. Right: mean fitting-and-query times with ordinary standard-deviation bars. The figure uses the original fixed configurations in Appendix C.

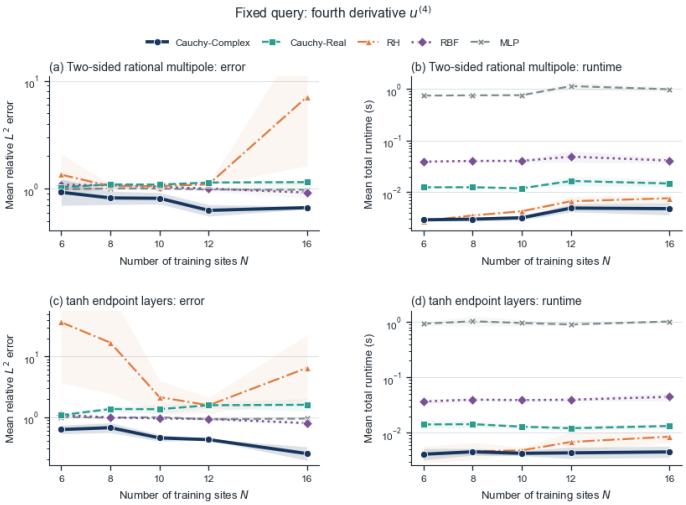


Figure 4: **Original fourth-derivative comparison across observation budgets.** Models fit values and slopes at $N \in \{6, 8, 10, 12, 16\}$ sites. Curves show five-seed arithmetic means; bands are $\overline{E}\exp(\pm \mathrm{std}(\log E))$, with the analogous construction for runtime. Total time includes fitting and all three derivative queries. Configurations remain fixed across the displayed budgets.

6 Discussion and Conclusion

Cauchy activations combine analytic operator assembly with direct coefficient recovery for low-dimensional linear PDEs. The controlled study gives lower errors than coefficient-only Adam on identical systems; Cauchy leads heat differential accuracy and sine leads plate accuracy under the common direct solver. The sparse-data study finds lower aggregate derivative errors on the specified near-singularity families. The theory separates finite-jet approximation inside a holomorphic neighborhood from recovery under finite constraints. These comparisons retain their stated development searches, manufactured problems, and solver budgets. Local pole dictionaries offer a route to richer domains, while fixed operators permit factorization reuse across data vectors.

REFERENCES

Anthony P. Austin, Peter Kravanja, and Lloyd N. Trefethen. Numerical algorithms based on analytic function values at roots of unity. *SIAM Journal on Numerical Analysis*, 52(4):1795–1821, 2014. doi: 10.1137/130931035.

Richard Baltensperger and Jean-Paul Berrut. The linear rational collocation method. *Journal of Computational and Applied Mathematics*, 134(1–2):243–258, 2001. doi: 10.1016/S0377-0427(00)00552-5.

Folkmar Bornemann. Accuracy and stability of computing high-order derivatives of analytic functions by Cauchy integrals. *Foundations of Computational Mathematics*, 11(1):1–63, 2011. doi: 10.1007/s10208-010-9075-z.

Ignacio Brevis, Ignacio Muga, David Pardo, Oscar Rodriguez, and Kristoffer G. van der Zee. Learning quantities of interest from parametric PDEs: An efficient neural-weighted minimal residual approach. *Computers & Mathematics with Applications*, 164:139–149, 2024. doi: 10.1016/j.camwa.2024.04.006.

Jingrun Chen, Xurong Chi, Weinan E, and Zhouwang Yang. Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method. *Journal of Machine Learning*, 1(3):268–298, 2022. doi: 10.4208/jml.220726.

Stefano Costa and Lloyd N. Trefethen. AAA-least squares rational approximation and solution of Laplace problems. In *European Congress of Mathematics*, pp. 511–534. EMS Press, 2023. doi: 10.4171/8ECM/16.

Wojciech Marian Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4278–4287, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/758a06618c69880a6cee5314ee42d52f-Abstract.html.

Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (PIELM)–a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020. doi: 10.1016/j.neucom.2019.12.099.

Abinand Gopal and Lloyd N. Trefethen. Solving Laplace problems with corner singularities via rational functions. *SIAM Journal on Numerical Analysis*, 57(5):2074–2094, 2019. doi: 10.1137/19M125947X.

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3):489–501, 2006. doi: 10.1016/j.neucom.2005.12.126.

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

Xin Li, Zhihong Xia, and Hongkun Zhang. Cauchy activation function and XNet. *Neural Networks*, 188:107375, 2025. doi: 10.1016/j.neunet.2025.107375.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Chunyang Liao, Deanna Needell, and Hayden Schaeffer. Cauchy random features for operator learning in Sobolev space, 2025. URL https://arxiv.org/abs/2503.00300.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5.

J. N. Lyness and C. B. Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, 1967. doi: 10.1137/0704019.

Yuji Nakatsukasa, Olivier Sète, and Lloyd N. Trefethen. The AAA algorithm for rational approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, 2018. doi: 10.1137/16M1106122.

J. Tinsley Oden and Serge Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & Mathematics with Applications*, 41(5–6):735–756, 2001. doi: 10.1016/S0898-1221(00)00317-5.

Thomas O'Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning. *Journal of Computational Physics*, 496:112555, 2024. doi: 10.1016/j.jcp.2023.112555.

Maziar Raissi, Paris Perdikaris, and George Em 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. doi: 10.1016/j.jcp.2018.10.045.

Robert Schaback and Holger Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica*, 15:543–639, 2006. doi: 10.1017/S0962492906270016.

Lloyd N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014. doi: 10.1137/130932132.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021. doi: 10.1137/20M1318043.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022. doi: 10.1016/j.cma.2022.114823.

Hong-Kun Zhang, Xin Li, Sikun Yang, and Zhihong Jeff Xia. CauchyNet: Compact and data-efficient learning using holomorphic activation functions. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=dpniDbdApf.

Reproducibility. Appendices B.1, C, D.1, D.2 and E specify the target functions, sampling, search spaces, selection, and executed configurations. The artifact contains the selected studies' complete records, histories, checkpoints, source hashes, and numerical checks. Original figures retain their supplied results. Complete proofs appear in Appendix A.

AI Use Statement Generative AI tools assisted with manuscript revision, mathematical exposition, code correction, numerical verification, and experiment execution. The authors take responsibility for the paper's text, claims, results, and code.

A Proofs

A.1 Proof of the exact derivative formula

Proof. For one coordinate,

$$\frac{\partial^{\alpha_j}}{\partial x_j^{\alpha_j}}(\xi_{k,j} - x_j)^{-1} = \alpha_j!(\xi_{k,j} - x_j)^{-(\alpha_j+1)}.$$

The product rule separates across coordinates because each factor depends on one coordinate. Hence

$$D^{\alpha}\phi_k(\boldsymbol{x}) = \prod_{j=1}^{d} \alpha_j!(\xi_{k,j} - x_j)^{-(\alpha_j+1)} = \alpha! \prod_{j=1}^{d} (\xi_{k,j} - x_j)^{-(\alpha_j+1)}.$$

The coordinatewise pole-avoidance assumption makes every factor finite. The sum is finite, so differentiation commutes with the sum and the real part. The constant contributes a column of ones at order zero and a zero column at positive orders. This gives Equations (2) and (3). □

A.2 Proof of shared-weight finite-jet quadrature

Proof. For $\boldsymbol{x} \in M$, the multivariate Cauchy formula gives

$$D^{\alpha} f(\boldsymbol{x}) = \frac{\alpha!}{(2\pi \mathrm{i})^d} \int_{\Gamma_1} \cdots \int_{\Gamma_d} f(\boldsymbol{\zeta}) \prod_{j=1}^{d} (\zeta_j - x_j)^{-(\alpha_j+1)} \, \mathrm{d}\zeta_d \cdots \mathrm{d}\zeta_1, \tag{18}$$

where $\Gamma_j = \{c_j + \rho_j e^{\mathrm{i}t_j} : 0 \le t_j < 2\pi\}$. Parameterize each contour by $\zeta_j(t_j) = c_j + \rho_j e^{\mathrm{i}t_j}$. Then Equation (18) is $\alpha!$ times the normalized integral over the d-torus of

$$G_{\alpha,\boldsymbol{x}}(\boldsymbol{t}) = f(\boldsymbol{\zeta}(\boldsymbol{t})) \prod_{j=1}^{d} \frac{\rho_j e^{\mathrm{i}t_j}}{(\zeta_j(t_j) - x_j)^{\alpha_j+1}}. \tag{19}$$

Fix a coordinate j and extend t_j to the complex plane. If $t_j = \tau_j + \mathrm{i}y_j$, then $|\rho_j e^{\mathrm{i}t_j}| = \rho_j e^{-y_j}$. The composition with f is analytic on a neighborhood of every point with $\rho_j e^{-y_j} \le R_j$. The denominator stays separated from every x_j with $|x_j - c_j| \le r_j$ when $\rho_j e^{-y_j} \ge s_j > r_j$. Thus the periodic integrand is analytic on a neighborhood of the closed strip

$$-\log(R_j/\rho_j) \le \operatorname{Im} t_j \le \log(\rho_j/s_j). \tag{20}$$

For a fixed α, compactness gives a uniform bound over this strip and over $\boldsymbol{x} \in M$.

The shifted trapezoidal rule for a periodic function analytic in a strip has an error bounded by the sum of the two geometric tails from the lower and upper strip boundaries (Trefethen & Weideman, 2014). Applied at the phased nodes in Equation (10), this gives

$$\sup_{\substack{\boldsymbol{x}\in M \\ \boldsymbol{t}_{-j}\in[0,2\pi]^{d-1}}} \left|(I_j - T_{j,m_j}) G_{\alpha,\boldsymbol{x}}(\boldsymbol{t}_{-j})\right| \le C_{\alpha,j} \left[\left(\frac{s_j}{\rho_j}\right)^{m_j} + \left(\frac{\rho_j}{R_j}\right)^{m_j} \right]. \tag{21}$$

Here $\boldsymbol{t}_{-j}$ collects all coordinates except t_j, I_j is normalized integration in coordinate j, and T_{j,m_j} is the m_j-point periodic trapezoidal rule. The constant absorbs the fixed-order factor, the strip supremum, and the geometric denominators.

For the tensor rule, insert one quadrature operator at a time. The difference telescopes as

$$I_1 \cdots I_d - T_{1,m_1} \cdots T_{d,m_d} = \sum_{j=1}^{d} T_{1,m_1} \cdots T_{j-1,m_{j-1}} (I_j - T_{j,m_j}) I_{j+1} \cdots I_d. \tag{22}$$

The remaining integration and quadrature operators are bounded on the same compact strips. Combining Equations (21) and (22) therefore yields a constant C_α, independent of $\boldsymbol{m}$, times the coordinate sum in Equation (12).

Evaluating the tensor trapezoidal rule in Equation (19) gives

$$\alpha! \sum_{\boldsymbol{k}} f(\boldsymbol{\xi}_{\boldsymbol{k}}) \prod_{j=1}^{d} \frac{\rho_j e^{\mathrm{i}t_{k_j,j}}}{m_j} (\xi_{k_j,j} - x_j)^{-(\alpha_j+1)}.$$

The coefficient multiplying the differentiated kernel is exactly $\gamma_{\boldsymbol{k}}$ in Equation (11) and is independent of α. Since the set $\{\alpha : |\alpha| \le a\}$ is finite, the construction holds for all of its members with the same coefficients.

For a function real-valued on M, taking the real part of the quadrature preserves or reduces the absolute error on M. □

A.3 Proof of the ridge and truncated-SVD output bounds

Proof. Let $H_\lambda = A^\top A + \lambda I$. Substituting $\boldsymbol{y} = A\boldsymbol{\theta}_m + \boldsymbol{e}$ gives

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_m &= H_\lambda^{-1} A^\top (A\boldsymbol{\theta}_m + \boldsymbol{e}) - \boldsymbol{\theta}_m \\
&= H_\lambda^{-1} A^\top \boldsymbol{e} + \left(H_\lambda^{-1} A^\top A - I\right) \boldsymbol{\theta}_m \\
&= H_\lambda^{-1} A^\top \boldsymbol{e} + H_\lambda^{-1} (A^\top A - H_\lambda) \boldsymbol{\theta}_m \\
&= H_\lambda^{-1} A^\top \boldsymbol{e} - \lambda H_\lambda^{-1} \boldsymbol{\theta}_m.
\end{aligned}$$

Multiplication by B and addition of $\boldsymbol{r}_B = B\boldsymbol{\theta}_m - \boldsymbol{z}$ give Equation (13). The triangle inequality and submultiplicativity give Equation (14).

For the truncated-SVD head,

$$\widehat{\boldsymbol{\theta}}_\tau - \boldsymbol{\theta}_m = A_\tau^\dagger \boldsymbol{e} + (P_\tau - I)\boldsymbol{\theta}_m.$$

Multiplication by B and addition of $\boldsymbol{r}_B$ give Equation (15); the triangle inequality gives Equation (16). □

A.4 Ambiguity within the exterior-pole space

Proposition A.1 (Finite-data ambiguity in the Cauchy space)**.** *Let m, n be positive integers, let $\xi_1, \dots, \xi_m$ be distinct complex numbers with $\operatorname{Im} \xi_k > 0$, and let $x_1, \dots, x_n$ be distinct real numbers. Let*

$$\mathcal{V}_m = \left\{ a_0 + \operatorname{Re} \sum_{k=1}^{m} \frac{c_k}{\xi_k - x} : \ a_0 \in \mathbb{R}, \ c_k \in \mathbb{C} \right\},$$

$$Q_\xi(x) = \prod_{k=1}^{m} (x - \xi_k)(x - \overline{\xi_k}), \quad S_n(x) = \prod_{i=1}^{n} (x - x_i)^2.$$

Let $T_n : \mathcal{V}_m \to \mathbb{R}^{2n}$ be $T_n v = (v(x_i), v'(x_i))_{i=1}^n$. Let $\mathbb{R}[x]_{\le r}$ denote real polynomials of degree at most r, with $\mathbb{R}[x]_{\le r} = \{0\}$ for $r < 0$. Then

$$\mathcal{V}_m = \{P/Q_\xi : \ P \in \mathbb{R}[x]_{\le 2m}\}, \qquad \dim \mathcal{V}_m = 2m + 1, \tag{23}$$

$$\ker T_n = \{S_n H/Q_\xi : \ H \in \mathbb{R}[x]_{\le 2m-2n}\}, \qquad \dim \ker T_n = \max\{2m + 1 - 2n, 0\}. \tag{24}$$

If $n \le m$, the function $h_n = S_n/Q_\xi$ belongs to $\ker T_n$ and satisfies

$$h_n''(x_i) = \frac{2 \prod_{j \ne i} (x_i - x_j)^2}{Q_\xi(x_i)} > 0 \qquad (i = 1, \dots, n). \tag{25}$$

Proof. The denominator Q_ξ has distinct conjugate roots and is strictly positive on the real line. Every realified feature has a real numerator over Q_ξ, with degree at most $2m$ after including the constant. Conversely, partial-fraction expansion of a real polynomial P of degree at most $2m$ divided by Q_ξ consists of a real constant and conjugate pairs of simple-pole terms. Each pair is the real part of a complex multiple of $(\xi_k - x)^{-1}$. This proves the first identity. Multiplication by Q_ξ is injective, so the dimension is $2m + 1$.

For $v = P/Q_\xi$, the conditions $v(x_i) = v'(x_i) = 0$ are equivalent to $P(x_i) = P'(x_i) = 0$, since $Q_\xi(x_i) > 0$. Distinct sites therefore imply that S_n divides P. Counting the coefficients of the remaining polynomial gives the kernel and its dimension, including the zero-kernel case $2n > 2m$. For $n \le m$, choose $H = 1$. At x_i, both S_n and S_n' vanish, and $S_n''(x_i) = 2 \prod_{j \ne i} (x_i - x_j)^2$. Differentiating the quotient twice gives Equation (25). □

For the selected $m = 12$, the exact nullities at $N = 6, 8, 10, 12, 16$ are $13, 9, 5, 1, 0$, respectively. Nonzero diagonal row and column scalings preserve these dimensions. Numerical truncation can discard further directions. Thus even an exactly interpolating coefficient vector can have a different unseen derivative from another member of the same trial space. The minimum-norm or regularized head is part of the continuation prior.

B Experiment specifications

B.1 Function definitions and numerical conventions

The function study uses four parameterized analytic families. Appendix D.1 specifies the parameter distributions and development/evaluation draws. The following fixed members of those families define the original function figures and the reference-function diagnostics in Appendix D.1. The figures use the original configurations in Appendix C; the diagnostics use the selected configurations below.

Reference functions. Write $L(x; c, w) = w/((x-c)^2 + w^2)$. On $[-1, 1]$, define

$$u_{\mathrm{R}}(x) = \sum_{k=0}^{11} (-1)^k e^{-k/4} L(x; 1.04 + 0.015k,\ 0.060 + 0.008k), \tag{26}$$

$$u_{\mathrm{B}}(x) = \sum_{k=0}^{7} e^{-k/4} \left[L(x; c_k, w_k) + 0.7(-1)^k L(x; -c_k, 1.2w_k) \right], \tag{27}$$

$$c_k = 1.04 + 0.018k, \qquad w_k = 0.065 + 0.010k,$$

$$u_{\log}(x) = \log\big((1.04 - x)^2 + 0.07^2\big), \tag{28}$$

$$u_{\tanh}(x) = \tanh\left(\frac{x - 1.03}{0.07}\right) - 0.6 \tanh\left(\frac{x + 1.04}{0.10}\right). \tag{29}$$

Each Lorentz term has poles $c \pm \mathrm{i}w$. The logarithmic branch agrees with the real logarithm on the interval and has branch points $1.04 \pm 0.07\mathrm{i}$. The tanh poles are $1.03 + 0.07\pi\mathrm{i}(k + 1/2)$ and $-1.04 + 0.10\pi\mathrm{i}(k + 1/2)$, $k \in \mathbb{Z}$. The respective nearest singularity moduli are approximately 1.04173, 1.04203, 1.04235, and 1.03585. The original radius-1.025 Cauchy configuration admits a common analytic neighborhood with $R = 1.03$ in Theorem 4.1. The selected radius 1.05 exceeds these four moduli, so its fitted predictions are assessed empirically. The algebraic identities in Proposition 4.2 apply to both geometries.

Reference derivatives are analytic. For a Lorentz term, $L^{(q)}(x; c, w) = \mathrm{Im}[-q!/(c - x + \mathrm{i}w)^{q+1}]$. For $q \geq 1$, the logarithmic derivative is $u_{\log}^{(q)}(x) = 2\,\mathrm{Re}[-(q-1)!/(1.04 - x + 0.07\mathrm{i})^q]$. Tanh derivatives through order four use the exact polynomials in $\tanh((x-c)/w)$ with chain-rule factors w^{-q}. The parameterized families use these same formulas with the drawn locations, widths, and amplitudes.

Sampling and queries. For a given seed, NumPy's default generator draws $v_i \sim \mathrm{Unif}[0, 1)$, $i = 0, \ldots, N-1$, and forms $x_i = 2(i + v_i)/N - 1$. The same generator randomly permutes the locations. All methods and targets share the seed-specific sites. Fitting uses orders $0, 1$; evaluation uses orders $2, 3, 4$ at $z_j = -1 + 2(j + 1/2)/2048$, $j = 0, \ldots, 2047$. The development query grid has the same midpoint construction with 1,024 points. Evaluation uses seeds $101, 211, 307, 401, 503$.

Linear-head normalization. For an unscaled basis function ψ_j, compute $s_j = (1024^{-1} \sum_{\ell=0}^{1023} \psi_j(g_\ell)^2)^{1/2}$ on $g_\ell = \cos(\pi(\ell + 1/2)/1024)$, and set $t_q = \max\{(N^{-1} \sum_i u^{(q)}(x_i)^2)^{1/2}, 10^{-14}\}$. For $q = 0, 1$, stack

$$A_{(q,i),j} = \frac{\psi_j^{(q)}(x_i)}{\sqrt{2N}\, t_q s_j}, \qquad y_{(q,i)} = \frac{u^{(q)}(x_i)}{\sqrt{2N}\, t_q}. \tag{30}$$

Query columns are divided by the same s_j. For $A = U\Sigma V^\top$, the fitted head is $V\,\mathrm{diag}(f_\lambda(\sigma_i))U^\top y$, with

$$f_0(\sigma) = \begin{cases} \sigma^{-1}, & \sigma > 10^{-12}\sigma_{\max}, \\ 0, & \text{otherwise}, \end{cases} \qquad f_\lambda(\sigma) = \frac{\sigma}{\sigma^2 + \lambda} \quad (\lambda > 0). \tag{31}$$

The positive-ridge filter acts on every singular value. Ridge penalizes the scaled coefficient vector, including its constant component.

Selected dictionaries. Cauchy-Complex uses $\xi_k = 1.05\exp(\mathrm{i}\pi(k+1/2)/12)$, $k = 0, \ldots, 11$, with both realified components. Cauchy-Real uses the real parts at $\xi_k = 1.1\exp(2\pi\mathrm{i}(k + 1/2)/64)$, $k = 0, \ldots, 63$. Chebyshev uses $T_0, \ldots, T_4$; Fourier uses $1, \cos(k\pi x), \sin(k\pi x)$ for $k = 1, 2$. Gaussian RBF uses 128 centers $c_j = -1 + 2j/127$, $j = 0, \ldots, 127$, and features $\exp[-(x - c_j)^2/(2(0.24)^2)]$. Each dictionary includes a constant. The stored head sizes are 25, 65, 5, 5, and 129, respectively; Cauchy-Real spans 33 independent functions. The selected ridge values appear in Table 5.

AAA-pole refitting. The adapter applies AAA's greedy support selection (Nakatsukasa et al., 2018) to the N observed values, with at most $\min\{4, N-1\}$ supports and stopping tolerance $10^{-8}\max\{1, \max_i |u_i|\}$. It obtains denominator roots from polynomial coefficients, retaining leading

terms above 10^{-14} times the largest coefficient magnitude. Finite roots at distance greater than 10^{-3} from $[-1, 1]$ enter the dictionary. Roots with imaginary magnitude at most 10^{-9} supply a real feature; upper-half-plane roots supply both realified components. The dictionary includes a constant, which also defines the fit when the filter leaves an empty pole set. A ridge-10^{-4} head then fits both observed orders using Equation (30). Thus the reported method uses AAA for pole selection and linear regression for its final value/slope predictions.

Neural model and optimization. The selected MLP has one width-32 tanh hidden layer, a linear output, and 97 parameters. PyTorch default linear-layer initialization uses the sampling seed; the network is then converted to float64. The training objective is $\sum_{q=0}^{1} N^{-1} \sum_i ((\widehat{u}^{(q)}(x_i) - u^{(q)}(x_i))/t_q)^2$. Adam takes 3,000 steps at learning rate 0.005 with its remaining settings at their defaults. One L-BFGS call uses `max_iter=80`, learning rate 1, gradient tolerance 10^{-10}, change tolerance 10^{-12}, history size 50, and strong-Wolfe line search. Automatic differentiation evaluates queries in batches of 512 and shares computation through order four.

Implementation and numerical checks. The implementation uses Python 3.13.2, NumPy 2.5.1, SciPy 1.18.0, and PyTorch 2.13.0 on macOS arm64, with one Torch and one BLAS thread per worker. Analytic target derivatives and dictionary derivatives have numerical cross-checks. The AAA implementation uses the full right singular basis for wide Loewner matrices, preserving their nullspace; a dedicated test covers this case. The result verifier checks selected configurations, complete run coverage, stored histories, and checkpoint residuals. Instrumented times in the training diagnostics describe execution under concurrent workers.

C Original figure configurations

The original supplied figures report fixed-configuration studies on the reference targets. The numerical tables report subsequent studies with development selection. Each figure keeps its original data, axes, and timing convention. The overview in Figure 1 is a schematic of shared coefficients and differentiated feature matrices; its order-$0/1/2$ illustration is one possible observation interface.

Function figures. Figures 3 and 4 use u_{B} and $u_{\tanh}$ from Equations (27) and (29), sampling seeds $\{101, 211, 307, 401, 503\}$, and $N \in \{6, 8, 10, 12, 16\}$. All methods fit values and first derivatives at the same stratified sites and evaluate orders two through four on a 2,048-point midpoint grid. Cauchy-Complex uses 16 upper-half-plane poles at radius 1.025, both realified components, a constant, and zero ridge, giving 33 coefficients. Cauchy-Real uses 64 full-circle real-part features at radius 1.4, a constant, and ridge 10^{-12}, giving 65 stored columns. Gaussian RBF uses 128 equally spaced centers on $[-1, 1]$, width 0.12, a constant, and zero ridge. These linear dictionaries use Equation (30) and the stated SVD cutoff. The original tanh MLP has architecture 1–64–1 and takes 1,500 Adam updates at rate 0.005, followed by at most 80 L-BFGS iterations. Function runtimes include fitting and all three derivative queries on the original machine. The figures retain their supplied five-seed means and dispersion bands.

PDE figure. Figure 2 uses the heat and plate equations in Section 5.1 and training point counts in Appendix D.2. The original heat CFC grid is 14×14, centered at $(1, \pi/2)$ with radii $(1.5, 2.5)$; the plate grid is 18×18, centered at $(\pi/2, \pi/2)$ with radii $(2.5, 2.5)$. Both use real-part tensor features, half-step phases, $a_0 = 0$, and ridge 10^{-10}. Their stored column counts are 196 and 324, with independent spans of at most 98 and 162. The original CFC objective is $\sum_b w_b \|r_b\|_2^2 + 10^{-10} \|\boldsymbol{\theta}\|_2^2$; the PINN uses $\sum_b w_b n_b^{-1} \|r_b\|_2^2$, with the same numerical weights $(1, 50, 50)$. The PINN architecture is 2–64–64–64–1, with 2,000 full-batch Adam updates at rate 0.002, float64 arithmetic, and one CPU thread. CFC/PINN evaluation grids are $70 \times 100/60 \times 90$ for heat and $80 \times 80/65 \times 65$ for plate. Times include construction, fitting, and evaluation on the original machine.

The original figure shows lower field and differential errors and shorter total runtimes for CFC under its recorded configurations. Tables 1 and 2 report the subsequent matched-loss and controlled-solver studies under their own protocols.

D MODEL SELECTION AND COMPLETE EVALUATION SUMMARIES

D.1 FUNCTION-STUDY DESIGN AND SELECTED CONFIGURATIONS

The function study uses the four parameterized families underlying Equations (26) to (29). For the multipoles, replace the locations by $c_0 + ak$, widths by $w_0 + bk$, and decay by $e^{-k/\tau}$. Right multipoles retain 12 alternating terms. Two-sided multipoles retain eight right terms and eight left terms, with left amplitude factor $\eta(-1)^k$ and left width multiplier γ. The remaining families are $\log((c-x)^2 + w^2)$ and $\tanh((x - c_R)/s_R) - \eta \tanh((x + c_L)/s_L)$.

Family	Uniform parameter ranges, drawn in the listed order
Right/two-sided	$c_0 \in [1.025, 1.075]$, $a \in [0.012, 0.020]$, $w_0 \in [0.045, 0.090]$, $b \in [0.006, 0.012]$, $\tau \in [3, 6]$; two-sided additionally $\eta \in [0.4, 0.85]$, $\gamma \in [1, 1.5]$
Logarithmic	$c \in [1.025, 1.085]$, $w \in [0.045, 0.10]$
Tanh	$c_R \in [1.015, 1.06]$, $s_R \in [0.05, 0.095]$, $c_L \in [1.02, 1.07]$, $s_L \in [0.075, 0.13]$, $\eta \in [0.4, 0.8]$

Table 4: **Target generator fixed before evaluation.** NumPy default generators use seeds 20260925 and 20260926 for development and evaluation, respectively. Families occur in the displayed order, with two development and three evaluation instances per family. Full-precision parameters are recorded in the artifact's target manifests.

Development uses sampling seeds 601 and 607, $N = 12$, and a 1,024-point midpoint query grid. Each configuration is fitted to orders zero and one; selection uses the geometric mean over the eight targets of their arithmetic mean E_4 over the two sampling seeds. Evaluation uses seeds 101, 211, 307, 401, and 503 and a 2,048-point query grid. Site sampling, column scaling, and block normalization follow Equation (30). Selection requires every fit of a candidate to succeed. The original configuration appears first in each candidate list, followed by the listed Cartesian products with duplicate configurations removed. Ties use candidate index.

Search spaces. Cauchy-Complex considers nonconstant column counts $\{16, 24, 32, 48\}$ and radii $\{1.005, 1.01, 1.025, 1.05, 1.1, 1.4\}$ at zero ridge (24 combinations), plus counts $\{24, 32\}$, radii $\{1.025, 1.1\}$, and ridge $\{10^{-12}, 10^{-8}\}$ (eight combinations). Cauchy-Real considers 32, 48, 64, or 96 full-circle columns at the same six radii with ridge 10^{-12}, plus counts $\{48, 64\}$ and radii $\{1.025, 1.1, 1.4, 1.8\}$ at zero ridge. Chebyshev degrees are $\{4, 6, 8, 10, 12, 16, 24, 32\}$ with ridge $\{0, 10^{-14}, 10^{-10}, 10^{-6}\}$. Fourier has the same set of nonconstant sine/cosine column counts and ridge $\{0, 10^{-12}, 10^{-8}, 10^{-4}\}$. RBF counts are $\{12, 24, 64, 128\}$, widths $\{0.06, 0.12, 0.24, 0.48\}$, and ridge $\{0, 10^{-8}\}$. AAA-pole refitting varies maximum supports $\{4, 6, 8, 12\}$, tolerances $\{10^{-8}, 10^{-12}\}$, and ridge $\{0, 10^{-12}, 10^{-8}, 10^{-4}\}$; its pole filter is unchanged. The MLP varies width $\{32, 64\}$, depth $\{1, 2\}$, Adam learning rate $\{0.001, 0.003, 0.005, 0.01\}$, and Adam steps $\{1500, 3000\}$, followed by at most 80 L-BFGS iterations with the settings in Appendix B.1. There are 224 candidates and 3,584 development fits in total.

The first candidates are Cauchy-Complex with 32 nonconstant columns, radius 1.025, and zero ridge; Cauchy-Real with 64 columns, radius 1.4, and ridge 10^{-12}; Chebyshev degree 32 with ridge 10^{-14}; Fourier with 16 nonconstant columns and ridge 10^{-8}; RBF with 128 centers, width 0.12, and zero ridge; AAA-pole refit with 12 maximum supports, tolerance 10^{-12}, and ridge 10^{-12}; and a 1–64–1 MLP with 1,500 Adam steps at 0.005 followed by the same 80-iteration L-BFGS limit. These configurations belong to the search spaces above and establish the saved candidate order.

Evaluation across derivative orders. The selected configurations produce 420 fits on new instances, and the original CFC configuration contributes 60 paired reference fits. Table 6 reports every queried derivative order under the same fits. Cauchy-Complex has the lowest aggregate error at orders two, three, and four. Selection uses E_4; the second- and third-derivative columns measure the outputs of those same selected models.

Reference-function diagnostics. A 700-fit diagnostic covers all seven selected methods, the four previously inspected functions in Appendix B.1, five site counts $\{6, 8, 10, 12, 16\}$, and five seeds. All

Method	Selected configuration, fixed for every evaluation target
Cauchy-Complex	24 nonconstant columns, radius 1.05, ridge 0
Cauchy-Real	64 real-part columns, radius 1.1, ridge 10^{-12}
Chebyshev	Degree 4, ridge 10^{-6}
Fourier	Four nonconstant columns (two frequencies), ridge 10^{-4}
Gaussian RBF	128 centers, width 0.24, ridge 10^{-8}
AAA-pole refit	Maximum supports 4, tolerance 10^{-8}, ridge 10^{-4}
tanh MLP	1–32–1; Adam 3,000 steps at 0.005; L-BFGS limit 80

Table 5: **Frozen function configurations.** All fixed dictionaries include a constant. Pole centers are zero with half-step phases. Each selected configuration is fixed before evaluation.

Method	E_2	E_3	E_4
Cauchy-Complex	**0.173**	**0.486**	**0.623**
Cauchy-Real	0.283	0.836	1.228
Chebyshev	0.788	0.996	1.000
Fourier	1.117	1.013	1.003
Gaussian RBF	0.642	0.901	0.974
AAA-pole refit	0.601	1.150	1.286
tanh MLP	0.479	0.890	1.043
CFC (original configuration)	0.325	0.665	0.804

Table 6: **All queried derivative orders on the 12 new instances.** Each entry is the geometric mean over instances of the five-seed mean relative error. Bold marks the lowest score among the seven selected methods. The original CFC configuration is a paired reference. All three derivatives use the same fitted coefficient vector or network.

1,180 evaluation and diagnostic fits succeed. The new-instance comparison evaluates fresh parameter draws, while these fixed reference functions provide training and observation-budget diagnostics. Table 7 compares the original and selected CFC configurations: selection improves the logarithmic target and increases error on the multipole and tanh reference functions. The paired new-instance results also contain both gains and regressions. Both comparisons use a configuration selected for an aggregate objective across development targets. Contour locations relative to the reference singularities are specified in Appendix B.1.

Reference function	Original E_4	Selected E_4	Selected/original
Right multipoles	0.779	0.931	1.194
Two-sided multipoles	0.629	0.833	1.324
Logarithmic	0.860	0.740	0.860
Tanh layers	0.424	0.560	1.321

Table 7: **CFC configurations on the four fixed reference functions.** Errors are arithmetic means over five sampling seeds at $N = 12$. Ratios below one favor the selected configuration and ratios above one favor the original configuration. These functions also define the training and observation-budget diagnostics retained in the artifact.

D.2 Matched-loss heat and plate studies

The PDE study uses the manufactured solutions in Section 5.1. Interior sites are tensor products of independently jittered axes: each axis point is drawn uniformly from the central half of its stratum. Heat uses 36×36 interior points, 80 equispaced initial points including endpoints, and 72 equispaced times on each spatial edge. Plate uses 30×30 interior points and 80 equispaced sites on each edge, with corners included on both adjacent edges. Value and Laplacian boundary constraints use the same 320 sites.

For each problem, CFC considers 27 configurations: heat has 10, 14, or 18 poles per axis; plate has 14, 18, or 22; both use common radius multipliers $\{0.9, 1, 1.2\}$ and ridge $\{0, 10^{-14}, 10^{-10}\}$.

Multipliers act on the original heat radii $(1.5, 2.5)$ and plate radii $(2.5, 2.5)$; centers and phase $1/2$ stay fixed. Zero-ridge solves retain singular values above $10^{-12}\sigma_{\max}$; positive-ridge solves use the ridge spectral filter. The PDE solver uses the original feature columns with each residual block scaled by $\sqrt{w_b/n_b}$.

The PINN compares learning rates $\{0.001, 0.002, 0.005\}$, each with 2,000 Adam updates and the same 2–64–64–64–1 architecture. Both methods use development sampling seeds 601 and 607. Independent validation points use NumPy default generators with seeds 1601 and 1607: uniform interior points and uniform edge/initial points, with the same block counts as training. Validation uses PDE forcing and boundary/initial data. Selection minimizes the geometric mean of the two validation weighted mean residuals. There are 108 CFC development fits and 12 PINN development fits, reflecting 27 geometry/ridge candidates and three learning-rate candidates per problem.

Heat selects an 18×18 CFC grid with multiplier 1.2 and ridge 10^{-14}; plate selects 22×22, multiplier 1.2, and zero ridge. Both PINNs select learning rate 0.001. These settings are frozen before the five evaluation seeds, producing ten CFC and ten PINN evaluations, all successful. The common field/query grids contain 60×90 points for heat and 65×65 for plate, including domain endpoints. Each block loss is its ordinary MSE, with weights $(1, 50, 50)$. Reported training residuals are the weighted block MSEs. The selected ridge, capacity, and optimizer settings specify the remaining parts of each method's fitting procedure.

The heat pole grid is centered at $(1, \pi/2)$ with selected radii $(1.8, 3.0)$; the plate grid is centered at $(\pi/2, \pi/2)$ with selected radii $(3.0, 3.0)$. Both use real-part tensor features, phase $1/2$, and $a_0 = 0$, giving 324 and 484 stored columns with spans of at most 162 and 242. The PINN maps inputs to $[-1, 1]^2$ and has 8,577 parameters. Every fit uses one CPU thread and float64 arithmetic.

Histories and checkpoints. The neural producer records loss before each Adam update. Histories label these states by completed updates $0, \ldots, 1999$ and append the loss after update 2,000. All final PINN weights are saved; validation and final training losses are evaluated from those networks. Exact-solution checks verify coordinate scaling and both PDE operators. Every recorded state, including transient increases, is retained in the artifact. CFC's direct solve contributes one terminal residual. These histories belong to the matched-loss study in Table 1.

Saved results. The artifact contains the target manifests, candidate configurations, frozen selections, per-case records, and final neural checkpoints. Table renderers read these records directly. Hash ledgers identify the saved results and execution sources for each study.

E Controlled feature and coefficient-solver comparison

Shared equations, points, and coordinates. The controls use the heat and plate problems in Section 5.1, with the same forcing, boundary/initial information, collocation counts, block-mean weights $(1, 50, 50)$, and evaluation grids. Development uses sampling seeds 601 and 607 and independent validation seeds 1601 and 1607, generated as in Appendix D.2. Evaluation uses the new seeds 809, 811, 821, 823, and 827. The new seeds vary the collocation points and, for random dictionaries, the fixed features. The target PDEs themselves remain the same manufactured problems.

Every method uses $p = 162$ real columns for heat and $p = 242$ for plate. For Cauchy, the 18×18 and 22×22 real-part tensor dictionaries contain identical conjugate-pair columns. We retain one representative per pair and multiply it by $\sqrt{2}$. If $[\Phi\ \Phi]$ denotes the paired columns, this compression preserves $[\Phi\ \Phi][\Phi\ \Phi]^\top = 2\Phi\Phi^\top$ and the symmetric minimum-norm coefficient coordinates. Each weighted design column is then divided by its Euclidean norm, with the same scale used for every output query. Ridge penalizes coefficients in these normalized coordinates. This shared normalization and the newly selected configurations distinguish the controls from the earlier PINN comparison.

Fixed dictionaries. For tanh, sine, RBF, and Chebyshev, map the physical rectangle affinely to $\boldsymbol{z} \in [-1, 1]^2$. Tanh and sine use $p - 1$ random features and a constant. For scale s, tanh uses $\tanh(s(\boldsymbol{w}_k^\top \boldsymbol{z} + b_k))$, with weights and biases uniform on $[-1, 1]$. Sine uses $\sin(s\boldsymbol{w}_k^\top \boldsymbol{z} + b_k)$, with the same weight law and independent phases uniform on $[-\pi, \pi]$. RBF uses $p - 1$ centers uniform on $[-1, 1]^2$, a constant, and $\exp(-\|\boldsymbol{z} - \boldsymbol{c}_k\|_2^2/(2s^2))$. The dictionary generator uses the sampling

seed plus 10,000; candidate scales share the same base draws. Chebyshev products $T_i(z_1)T_j(z_2)$ are sorted by $(\max\{i,j\}, i+j, i, j)$, retaining the first p. All physical-coordinate derivatives are evaluated analytically.

Development selection. Cauchy considers radius multipliers $\{0.8, 0.9, 1, 1.1, 1.2, 1.4, 1.6, 2, 2.5\}$ on the base radii $(1.5, 2.5)$ for heat and $(2.5, 2.5)$ for plate, with the preceding centers and phase $1/2$. Tanh, sine, and RBF consider scales $\{1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8, 16\}$. Every dictionary considers ridge $\{0, 10^{-14}, 10^{-10}\}$. This gives 27 candidates per activation dictionary and three for the fixed Chebyshev basis: 444 direct development fits over both PDEs and two seeds. Selection minimizes the geometric mean of the terminal validation residuals, breaking ties by candidate index. The common solver uses the spectral filter in Equation (7), with $\tau = 10^{-12}\sigma_{\max}$ at zero ridge.

	Heat		Plate	
Method	Scale	Ridge	Scale	Ridge
Cauchy	2	0	1.2	0
Fixed tanh	1	0	1	0
Fixed sine	2	0	8	0
Gaussian RBF	1	0	0.5	0
Chebyshev	—	0	—	0

Table 8: **Selected controlled-study configurations.** Cauchy's scale is its radius multiplier; RBF's is its width in normalized coordinates; tanh and sine scales multiply the fixed input weights as specified above. Every setting is fixed before the new-seed evaluation.

Coefficient-only Adam. The selected Cauchy matrix, right-hand side, column scales, and ridge define a common objective for SVD and Adam. Adam starts from zero coefficients and uses 20,000 full-batch updates, moment factors $(0.9, 0.999)$, and $\epsilon = 10^{-8}$. The constant learning-rate candidates are $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, giving 20 development fits. Terminal validation residuals select rate 1 for both PDEs. The gradient is evaluated directly as $2A^\top(A\boldsymbol{\theta} - \boldsymbol{y}) + 2\lambda\boldsymbol{\theta}$ from the current residual. Adam updates the full fixed coefficient vector, while the zero-ridge SVD applies its declared truncation. This comparison is conditional on the dictionary selected for direct fitting and the specified constant-rate Adam search.

The frozen evaluation contains 50 direct fits and ten Adam fits; all 464 development fits and 60 evaluations complete successfully. Residuals are saved every 100 updates, including step zero, and coefficients are saved after 0, 2,000, 10,000, and 20,000 updates. The primary Adam endpoint is update 20,000. The artifact also records the first logged crossing of training residuals 10^{-4}, 10^{-8}, and 10^{-12}, at this 100-update resolution.

Timing and seedwise accuracy. Execution is serial with one CPU thread and float64 arithmetic. Fit time includes feature construction, operator assembly, column scaling, and coefficient fitting; independent validation, output-grid evaluation, and file writing are separate. Evaluation order rotates across seeds. The times in Table 2 describe completion of the stated solves and Adam budget on the local host. The artifact's `all_evaluations.csv` retains every seed-specific field error, differential error, and fit time underlying the medians.

Numerical verification. The artifact checks all mixed derivatives through total order four against automatic differentiation and verifies the Adam implementation against PyTorch. Replaying all 60 saved coefficient states reproduces training residuals, validation residuals, and field/differential errors. Hashes verify identical matrices in all ten Cauchy solver pairs. The plate RBF fits have coefficient norms of order 10^8, and their output-scaled stationarity defects range from 1.2×10^{-8} to 2.7×10^{-8}. The corresponding defects scaled by $\|A\|_2^2\|\boldsymbol{\theta}\|_2 + \|A\|_2\|\boldsymbol{y}\|_2$ are below 2.5×10^{-16}. Both diagnostics are retained alongside the finite-precision results. Source snapshots, selection timestamps, all candidate records, and coefficient checkpoints accompany the study.