CORRECTING TRAINED PINNS FROM RESIDUALS: OPERATOR-KERNEL INFERENCE OF THE ERROR FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

A trained physics-informed neural network (PINN) retains computable PDE residuals and boundary and initial-condition defects that contain information about its remaining solution error. We use these defects to reconstruct a global correction while keeping the network fixed. The governing equation turns them into operator observations of the unknown error, exactly for linear PDEs and through a frozen linearization for nonlinear PDEs. Applying the observation operators to a kernel produces a finite reconstruction system whose hyperparameters are fitted from the defects. Regularized linear solves then recover the correction without observing the true error or retraining the PINN. Our analysis characterizes the conditional RKHS estimator and separates finite-data reconstruction from nonlinear linearization. Experiments across canonical PDEs, held-out physical instances and irregular geometry show reductions in solution error. On a nonlinear variable-coefficient problem with directly specified forcing, the complete procedure substantially improves the trained field. For this problem, it also achieves lower error than neural error-field fitting and continued training given larger compute allowances. Budget comparisons reveal problem-dependent tradeoffs, while resolved Newton references identify reconstruction as the larger discrepancy on the studied Burgers cases. These results connect observable equation defects to a usable correction of an existing neural field.

1 INTRODUCTION

A physics-informed neural network (PINN) represents a PDE solution by a neural field u_θ, trained to satisfy the equation and its conditions at sampled collocation points (Raissi et al., 2019). After training, the field and its derivatives remain available, even when optimization has left substantial solution error, as documented in PINN failure studies (Krishnapriyan et al., 2021). The governing equation therefore provides information that can still be used to improve the trained field. We ask how to turn this information into a global correction while keeping the network fixed.

For $N(u) = f$ with boundary data $B(u) = g$, the error $e := u - u_\theta$ depends on the unknown solution, whereas the residual $r := N(u_\theta) - f$ and boundary defect $r_\partial := B(u_\theta) - g$ are computable. Their magnitudes alone do not determine the error: the equation may be nearly satisfied where the error is large. The equation instead relates these fields through an operator. For a linear PDE, $Le = -r$ and $Be = -r_\partial$ hold exactly. For a nonlinear PDE, linearization gives $N'(u_\theta)e = -r + \eta$, with a Taylor remainder η; the frozen field supplies the coefficients of $N'(u_\theta)$. Residual and condition-defect evaluations thus provide exact or approximate *operator observations* of the unknown error.

We reconstruct the error from these observations with a kernel model. Automatic differentiation extracts the operator, and its action on both arguments of a kernel produces a Gram matrix coupling scattered interior, boundary and initial observations. A small set of kernel hyperparameters is fitted from the defects, and regularized linear systems recover the correction coefficients. The nonlinear procedure keeps the same frozen operator through its two regularization stages. Adding the inferred field gives $\widetilde{u} = u_\theta + \widehat{e}$, without changing the network or observing its true error.

Classical defect correction solves an auxiliary error equation on a numerical discretization (Krasowski et al., 2026), while neural error-field methods fit a second network (Sun & Zhang, 2026). Operator-conditioned kernels provide another representation of that equation (Raissi et al., 2018; Chen et al., 2021). Our procedure uses this representation to recover the error of an existing PINN

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.

from its observable defects. The kernel fit determines the reconstruction geometry; conditional on those parameters, the field coefficients follow from a regularized RKHS problem.

We characterize the resulting estimator and separate finite-data reconstruction from nonlinear linearization. The experiments test the complete procedure across nine canonical PDEs, held-out physical instances and an irregular domain. A two-dimensional variable-coefficient nonlinear problem with directly specified forcing provides a further test: against an independent numerical reference, the complete two-stage method reduces error by approximately two orders of magnitude and remains more accurate than neural error-field fitting and continued training even when those alternatives receive four times its additional compute. Paired budget comparisons identify problem-dependent tradeoffs, and resolved Newton references locate the larger discrepancy on the studied Burgers cases in reconstruction. The construction, its conditional analysis and these controlled measurements show how observable defects can support a global correction of an already trained neural field.

2 Related Work

Neural PDE approximations and training. Neural solvers include collocation-based residual fitting (Raissi et al., 2019; Sirignano & Spiliopoulos, 2018), variational objectives (E & Yu, 2018; Kharazmi et al., 2021) and domain-decomposed formulations (Jagtap et al., 2020; Jagtap & Karniadakis, 2020). Reviews organize these developments within physics-informed learning (Karniadakis et al., 2021; Cuomo et al., 2022). Their attainable accuracy also depends on optimization: gradient imbalance, unequal component convergence and loss conditioning have motivated changes to training (Wang et al., 2021; 2022; Rathore et al., 2024). Adaptive loss weights (McClenny & Braga-Neto, 2023), causal weighting (Wang et al., 2024) and point selection (Wu et al., 2023; Lau et al., 2024) act on that training process. Our correction starts from the resulting frozen field and uses a separate set of operator observations.

Residual correction and refinement. Classical a posteriori analysis and defect correction use an approximation's residual to recover information about its error (Ainsworth & Oden, 2000; Becker & Rannacher, 2001). Lagaris et al. (1998) already describe neural residual corrections for linear differential equations. Recent approaches discretize a PINN error equation by finite differences (Krasowski et al., 2026), fit an auxiliary error network (Sun & Zhang, 2026), add successive neural residual stages (Wang & Lai, 2024) or simulate a structure-preserving defect law at inference time (Fan et al., 2026). Learned denoising-based refinement also improves neural PDE rollouts (Lippe et al., 2023). Cao et al. (2023) and Jha (2024) solve linearized residual problems for externally supplied neural-operator predictions. Cao's ideal correction is a single Newton step with a local quadratic reduction under stated assumptions. PhysicsCorrect applies linearized discrete-residual corrections during temporal rollouts (Huang & Perdikaris, 2026); neural-initialized Newton instead uses a surrogate prediction to start nonlinear finite-element iterations (Taghikhani et al., 2026). The estimator studied here reconstructs a continuous error field from scattered operator observations of a frozen PINN. Its finite-data reconstruction error is distinct from the accuracy of an ideal Newton step.

Operator-conditioned kernels. Kernel collocation represents differential and boundary data through kernel-generated trial spaces (Fasshauer, 1997; Wendland, 2005). Gaussian-process conditioning provides a related probabilistic construction (Cockayne et al., 2016; Raissi et al., 2018), while operator-adapted kernels also arise in Bayesian numerical homogenization and multiresolution information games (Owhadi, 2015; 2017). For nonlinear PDEs, Chen et al. (2021) solve successive linearizations in a Gauss–Newton iteration. Batlle et al. (2025) analyse kernel/GP errors and give a linearize-then-optimize scheme in Algorithm 2.2: each linearized problem is recovered by a minimum-norm kernel solve. Their formal subproblems impose exact linearized constraints; their numerical implementation uses adaptive diagonal nuggets (Batlle et al., 2025, Appendix C). Our post-hoc procedure keeps the linearization and solution-space mean fixed at u_θ, fits a block-scaled regularization level, then applies the prescribed nonlinear second stage. Pförtner et al. (2024) give the arbitrary-mean operator-GP formula for linear PDEs, which includes this choice of centre. For linear equations, our error-coordinate construction is an affine reparameterization of that solution-coordinate estimator; Appendix H gives the direct-solver comparison.

Error assessment. Stability-based PINN analyses relate solution error to training and sampling errors (Mishra & Molinaro, 2023; De Ryck et al., 2024). Anwer et al. (2026) derive pointwise intervals for compatible mesh-based numerical PINN fields and a signed correction toward the discrete target in their square linear setting. Huseynov et al. (2026) give ODE error bounds under localized monotonicity and growth assumptions. Bayesian PINNs model a posterior over solutions (Yang et al., 2021), and model-discrepancy formulations represent departures from a computer model (Kennedy & O'Hagan, 2001). Our conditional analysis concerns the finite operator-kernel estimator and separates its reconstruction discrepancy from nonlinear linearization; the experiments measure both against resolved references.

3 Operator-Kernel Error Correction

3.1 From a Frozen PINN to an Error Equation

We are given a PDE $N(u) = f$ on a bounded $\Omega \subset \mathbb{R}^d$ with conditions $B_j(u) = g_j$ on prescribed faces, $j = 1, \dots, J$, and a trained network u_θ. The correction needs only a differentiable, callable u_θ and the known equation and data; it uses no optimizer state, training trajectory or collocation set, and leaves the network parameters fixed. Time-dependent problems are treated as initial–boundary-value problems, so no terminal-time information enters.

Write

$$e := u - u_\theta, \qquad r := N(u_\theta) - f, \qquad r_{\partial,j} := B_j(u_\theta) - g_j, \tag{1}$$

where only the last two can be evaluated. Substituting $u = u_\theta + e$ into the governing equation and into the boundary and initial conditions relates them:

$$A_\theta\, e = -r + \eta, \qquad B_j e = -r_{\partial,j}, \qquad (A_\theta, \eta) = \begin{cases} (L,\, 0), & N = L \text{ linear}, \\ (N'(u_\theta),\, -R(e)), & N \text{ nonlinear}, \end{cases} \tag{2}$$

Here $R(e) = N(u_\theta + e) - N(u_\theta) - N'(u_\theta)e$ is the Taylor remainder, quadratic in e under the local smoothness conditions of Proposition 2. The condition operators B_j are linear. For a linear operator η vanishes: the residual is the exact forcing of the error equation, whatever accuracy training reached, and no smallness of $\|e\|$ is needed. For a nonlinear operator, $A_\theta = N'(u_\theta)$ is the linearization around the frozen network and $\eta = -R(e)$ is the negative of its Taylor remainder.

For viscous Burgers, $N(u) = u_t + uu_x - \nu u_{xx}$, linearization gives

$$A_\theta v = v_t + u_\theta v_x + (u_\theta)_x\, v - \nu v_{xx}, \tag{3}$$

so the frozen network supplies both the right-hand side r and the spatially varying coefficients u_θ and $(u_\theta)_x$ of the operator acting on the error; these coefficient fields are recovered by Jacobian–vector products, which keeps one implementation valid for every operator in the suite (Appendix B).

3.2 Residual Evaluations as Operator Data

Equation 2 holds at every point of Ω, but only finitely many evaluations are affordable. For a single interior site x_i define the functional

$$\lambda_i(v) := (A_\theta v)(x_i), \qquad \text{so that} \qquad \lambda_i(e) = -r(x_i) + \eta(x_i), \tag{4}$$

which reduces to $\lambda_i(e) = -r(x_i)$ when N is linear. Each residual value is therefore, up to sign, a linear functional of the entire error field determined by the PDE operator, up to the sampled remainder $\eta(x_i)$ in the nonlinear case. A second-order operator ties together the value and the first and second derivatives of e at x_i; the kernel model couples this local information across the domain.

Boundary and initial conditions supply functionals of the same kind: for a site $z_{j,q}$ on the face carrying condition j,

$$\lambda^{\partial}_{j,q}(v) := (B_j v)(z_{j,q}), \qquad \lambda^{\partial}_{j,q}(e) = -r_{\partial,j}(z_{j,q}), \tag{5}$$

with the operator the physical condition imposes: a trace for Dirichlet data, a normal derivative for Neumann, a trace on $t = 0$ for a prescribed initial value, and ∂_t on $t = 0$ for an initial velocity. Each condition enters the observations through its own operator.

Collecting m interior sites $X = \{x_i\}$ and, for each group j, face sites X_j, gives the full observation operator and the vector of observed values,

$$\Lambda = \big(A_\theta|_{x_1}, \ldots, A_\theta|_{x_m},\, B_1|_{X_1}, \ldots, B_J|_{X_J}\big), \qquad y = -\big[\, r(X);\; r_{\partial,1}(X_1);\; \ldots;\; r_{\partial,J}(X_J)\,\big], \tag{6}$$

the first m entries interior and the rest boundary or initial. Writing $\boldsymbol{\eta}_\Lambda := [\,\eta(X);\, 0;\, \ldots;\, 0\,] \in \mathbb{R}^n$ for the remainder sampled at the interior sites, with n the total number of observations, the finite observation equation reads

$$\Lambda e = y + \boldsymbol{\eta}_\Lambda, \tag{7}$$

in which only the interior entries of $\boldsymbol{\eta}_\Lambda$ can be nonzero and $\boldsymbol{\eta}_\Lambda = 0$ for a linear PDE.

3.3 Operator-Kernel Reconstruction

Kernel methods recover a function from finitely many linear measurements. With a kernel k and RKHS $\mathcal{H}_k$ (Aronszajn, 1950), the native-space error bounds assume $e \in \mathcal{H}_k$; the same formulas arise as the conditional mean and covariance of an auxiliary $e \sim \mathcal{GP}(0, k)$ observed with covariance Σ, which is the same estimator under a different assumption, since sample paths of $\mathcal{GP}(0, k)$ generally do not lie in $\mathcal{H}_k$ (Appendix A.1). Every functional in Λ must be bounded on $\mathcal{H}_k$, which for the second-order interior operators here follows from a mixed-derivative condition on the kernel (Appendix B.4).

Each functional $\lambda_i \in \Lambda$ induces a *representer*, the kernel with that functional applied to its second argument,

$$\phi_i(x) := \lambda_i' k(x, \cdot), \tag{8}$$

and the reconstruction is a finite combination of these,

$$\widehat{e}(x) = \sum_{i=1}^{n} c_i\, \phi_i(x). \tag{9}$$

Every observation contributes one basis function whose shape its operator fixes: an interior residual gives the kernel differentiated by the PDE operator, a Dirichlet observation an undifferentiated slice, a Neumann or initial-velocity observation a first derivative. Residual and defect values enter only through the weights; Appendix B.3 specifies site sampling and the observation blocks.

Fitting the expansion to the observations under the regularization Σ gives the operator Gram matrix and the vector of representers evaluated at a query point,

$$[K_{\Lambda\Lambda}]_{ij} = \lambda_i(\phi_j) = \lambda_i \lambda_j' k, \qquad [k_\Lambda(x)]_i = \phi_i(x), \qquad M = K_{\Lambda\Lambda} + \Sigma, \tag{10}$$

where the prime marks action on the second kernel argument and $\Sigma \succ 0$ is the regularization specified in Section 3.5. The coefficients solve

$$Mc = y, \tag{11}$$

by a factorization rather than by forming M^{-1}, and the corrected field is

$$\widehat{e}(x) = k_\Lambda(x)^\top c, \qquad \widetilde{u} = u_\theta + \widehat{e}. \tag{12}$$

A worked Poisson construction. For $-u'' = f$ on $(0, 1)$ with Dirichlet endpoints, the network supplies $r = -u_\theta'' - f$. The interior data specify $-e''(x_i) = -r(x_i)$, while the endpoint defects specify $e(0) = g_0 - u_\theta(0)$ and $e(1) = g_1 - u_\theta(1)$. The expansion therefore takes the explicit form

$$\widehat{e}(x) = -\sum_{i=1}^{m} c_i\, \partial_z^2 k(x, z)\big|_{z=x_i} + c_0^{\partial} k(x, 0) + c_1^{\partial} k(x, 1).$$

The coefficients are solved jointly from equation 11. An interior–interior Gram entry uses $\partial_x^2 \partial_z^2 k(x_i, x_j)$, whereas an interior–boundary entry differentiates one argument and evaluates the other at an endpoint. Even a zero boundary defect supplies information, penalizing departures from zero in the reconstructed boundary error. The residual supplies derivative observations and the boundary supplies value observations; the shared system combines them into one continuous field that can be evaluated between the observation sites.

3.4 What the Reconstruction Computes

The finite expansion also identifies which reconstruction the system selects among all fields in $\mathcal{H}_k$.

Proposition 1 (Regularized minimum-norm correction)**.** *Under (A1)–(A3) of Appendix A and conditional on the fitted kernel hyperparameters, $\widehat{e}$ of equation 12 is the unique minimizer over $v \in \mathcal{H}_k$ of*

$$\mathcal{J}(v) \;=\; \|v\|^2_{\mathcal{H}_k} \;+\; \|\Lambda v - y\|^2_{\Sigma^{-1}}, \qquad \|z\|^2_{\Sigma^{-1}} := z^\top \Sigma^{-1} z. \tag{13}$$

If $\Sigma = \varepsilon D$ with fixed $D \succ 0$ and the observations are compatible, $y \in \operatorname{ran}\Lambda$, then as $\varepsilon \to 0$ it converges in $\mathcal{H}_k$ to the minimum-RKHS-norm solution of $\Lambda v = y$.

The two terms balance kernel norm and misfit to the observed error equation, with each observation block on its own scale. Once the kernel and regularization are fitted, the single system equation 11 determines the coefficients of a field on all of Ω. Positive regularization permits observation misfit; its zero limit recovers exact interpolation of compatible data.

For nonlinear problems the observations are exact for a different target: the Newton correction δ solves $A_\theta \delta = -r$, $B_j \delta = -r_{\partial,j}$, so $\Lambda\delta = y$ holds exactly while $\Lambda e = y + \boldsymbol{\eta}_\Lambda$, and the gap to the true error splits as

$$e - \widehat{e} = (e - \delta) + (\delta - \widehat{e}). \tag{14}$$

Proposition 2 (Conditional approximate-Newton decomposition)**.** *In the spaces X, Y of Appendix A.1, assume that N is Fréchet differentiable on a convex neighbourhood of u_θ and u with L_N-Lipschitz derivative; that the linearized boundary-value operator $\mathcal{A}_\theta v = (A_\theta v, B_1 v, \ldots, B_J v)$ maps X bijectively onto the admissible data space $\mathcal{Z}_{\mathrm{adm}}$, which contains the defect data, with $\|\mathcal{A}_\theta^{-1}\| \le C_A$; and that the representers ϕ_i lie in X. Then $\|e - \delta\|_X \le \frac{C_A L_N}{2}\|e\|^2_X$, and therefore*

$$\|e - \widehat{e}\|_X \;\le\; \underbrace{\|\delta - \widehat{e}\|_X}_{\textit{finite-data reconstruction}} \;+\; \underbrace{\frac{C_A L_N}{2}\|e\|^2_X}_{\textit{nonlinear linearization}}. \tag{15}$$

The first term is the error of rebuilding δ from finitely many scattered measurements; the second is the linearization error of one Newton step, quadratic in $\|e\|_X$. They cannot be separated without the reference solution, and bounded invertibility of the linearized problem is assumed rather than proved for Burgers and Allen–Cahn (Appendix A.5), so the proposition is a conditional approximate-Newton interpretation. The factorization of M also returns the auxiliary model's conditional covariance, which Appendix A identifies as a regularized operator power function: it measures the directions of the assumed RKHS left unresolved by the regularized observations, not a calibrated confidence interval.

3.5 Regularization, Hyperparameters, and Computation

The correction is fixed once the block weighting and the kernel scales are; both enter equation 13, in which Σ penalizes mismatch per block. An interior observation applies a second-order operator to each kernel argument, bringing in high inverse powers of the lengthscales, while a Dirichlet trace applies none, so block prior variances differ by orders of magnitude and one unscaled diagonal large enough for the interior would over-regularize the boundary. Following the operator-scale principle of adaptive nuggets (Chen et al., 2021; Batlle et al., 2025), we use

$$\Sigma = s^2\big(\sigma^2_{\mathrm{rel}} + \tau\big)D, \tag{16}$$

with D diagonal, each entry the mean amplitude-free prior variance of the observation's block divided by its multiplicity weight (Appendix B.5), and τ the numerical jitter of Appendix B.6, so that σ_{rel} is a relative level comparable across interior, boundary and initial-condition observations. Σ is regularization and numerical stabilization rather than a model of measurement noise, and for a nonlinear PDE it is also used to tolerate the aggregate mismatch $\boldsymbol{\eta}_\Lambda$ through an amplitude-heuristic regularization update (Appendix B.6); Appendix D reports the effect of replacing block-scaled regularization by a single global diagonal.

Each coordinate-wise lengthscale sets how far the information in one operator observation reaches, and σ_{rel} sets how closely the field is fitted to the observations. The scalar amplitude is profiled out analytically, and the lengthscales and σ_{rel} are selected on the profiled marginal objective by

Algorithm 1 Post-hoc operator-kernel correction of a frozen PINN

Input: trained u_θ; operators N, $\{B_j\}$; data f, $\{g_j\}$; interior sites X; face sites $\{X_j\}$; kernel family k; fixed evaluation sites for Stage 2
Evaluate the interior residual $r(X)$ and the boundary and initial defects $r_{\partial,j}(X_j)$ by automatic differentiation, and set $A_\theta = N'(u_\theta)$ by Jacobian–vector products if N is nonlinear, $A_\theta = L$ otherwise
Form the observation functionals Λ and targets y of equation 6, the operator-shaped representers ϕ_i, and the Gram matrix $K_{\Lambda\Lambda}$ of equation 10
Form the block-scaled Σ of equation 16 and fit the lengthscales and σ_{rel} by the multistart profiled fit of Appendix B {Stage 1}
Solve $(K_{\Lambda\Lambda} + \Sigma)\, c_1 = y$ and set $\widehat{e}_1(x) = k_\Lambda(x)^\top c_1$
if N is nonlinear **then**
    Estimate C_N on the first $\min(64, m)$ interior sites from the fixed second-order Taylor basis $\{1\} \cup \{x_a - p_a\} \cup \{(x_i - p_i)(x_j - p_j)\}_{i \le j}$, take $\|\widehat{e}_1\|_\infty$ over the evaluation grid, and set σ_{after} by the amplitude heuristic equation 37 {Stage 2}
    Replace σ_{rel} by σ_{after} in $\widetilde{M}$, holding the lengthscales, amplitude-free kernel, D, jitter and A_θ fixed; this updates *every* block, interior and boundary alike
    Re-profile $\widehat{s}^2$ by equation 36, update the common amplitude of $K_{\Lambda\Lambda}$, Σ and k_Λ, and solve $(K_{\Lambda\Lambda} + \Sigma)\, c = y$ once more
else
    $c \leftarrow c_1$
end if
Return: $\widehat{e}(x) = k_\Lambda(x)^\top c$ and $\widetilde{u}(x) = u_\theta(x) + \widehat{e}(x)$

multi-start L-BFGS-B (Appendix B), using only the frozen network, the known PDE data and the observations computed from them.

For n observations, the dense Gram matrix requires $O(n^2)$ memory and a dense factorization $O(n^3)$ time. Each evaluation of the marginal objective during hyperparameter fitting requires a factorization, followed by the conditioning solves of Algorithm 1; Appendix B reports that the fit dominates the wall-clock.

4 Experiments

4.1 Evaluation Setup

We evaluate three aspects of the correction: error reduction across problems and physical instances, accuracy gained from additional computation, and recovery of the spatial error field. A nonlinear problem with directly specified forcing provides an independent numerical-reference test; Newton references and component ablations separate the sources of the remaining error.

The canonical suite comprises Poisson in one and two dimensions, reaction–diffusion, heat, wave, Helmholtz, convection, Allen–Cahn and viscous Burgers. Each configuration has seven independent network initializations and point draws; all arms start from the same cached 4×64 tanh PINN within a run. Continued training updates that network; correction arms keep it fixed. The canonical Helmholtz operator is negative definite under its Dirichlet conditions (Appendix C.1). The held-out study instead treats each physical instance as the statistical unit. We report relative L^2 errors as arithmetic mean $\pm$ sample standard deviation and multiplicative comparisons as paired geometric means with 95% log-ratio t intervals. Intervals containing one leave the direction unresolved. Appendix C specifies the measures and pairing for each study.

Correction fitting and checkpoint selection use the frozen field and known PDE data. The canonical training budgets were calibrated once on seed 0 using reference error. The new nonlinear problem uses a fixed $40{,}000$-step budget without reference-based calibration, followed by the canonical kernel, observation mechanism and fitting rules, including Stage 2. Appendix B gives the implementation, and Appendix D includes the calibration-seed sensitivity analysis.

Table 1: Relative L^2 error (mean $\pm$ sample standard deviation over seven independent runs) on the canonical benchmarks, starting from the same pretrained PINNs. Neural alternatives receive the original kernel fit-and-readout budget, excluding observation construction. Bold marks the smallest mean among the three additional-compute arms. Paired comparisons are in Appendix D; Appendix I compares alternatives under the complete correction cost.

Benchmark	Raw PINN	Operator-kernel	Neural error field	Continued training
Poisson 1D	$(4.42\pm4.64)\times10^{-3}$	$\mathbf{(4.84\pm5.17)\times10^{-10}}$	$(1.96\pm1.58)\times10^{-3}$	$(1.89\pm1.02)\times10^{-4}$
React.–diff. 1D	$(4.98\pm2.62)\times10^{-3}$	$\mathbf{(1.96\pm0.58)\times10^{-8}}$	$(5.14\pm2.61)\times10^{-3}$	$(2.40\pm0.86)\times10^{-3}$
Heat 1D	$(4.42\pm1.94)\times10^{-3}$	$\mathbf{(1.22\pm0.86)\times10^{-7}}$	$(3.49\pm0.44)\times10^{-3}$	$(2.18\pm0.20)\times10^{-3}$
Wave 1D	$(1.40\pm1.33)\times10^{-2}$	$\mathbf{(3.92\pm2.57)\times10^{-6}}$	$(8.60\pm2.02)\times10^{-3}$	$(6.55\pm1.26)\times10^{-3}$
Poisson 2D	$(7.78\pm1.48)\times10^{-3}$	$\mathbf{(8.54\pm2.79)\times10^{-4}}$	$(7.43\pm1.23)\times10^{-3}$	$(7.01\pm1.24)\times10^{-3}$
Helmholtz 2D	$(1.86\pm0.92)\times10^{-2}$	$\mathbf{(4.14\pm1.54)\times10^{-3}}$	$(1.42\pm0.58)\times10^{-2}$	$(1.01\pm0.55)\times10^{-2}$
Convection 1D	$(8.72\pm5.04)\times10^{-3}$	$(3.84\pm2.46)\times10^{-3}$	$(5.49\pm2.03)\times10^{-3}$	$\mathbf{(3.50\pm2.58)\times10^{-3}}$
Allen–Cahn 1D	$(1.29\pm0.43)\times10^{-2}$	$\mathbf{(2.04\pm1.41)\times10^{-3}}$	$(1.15\pm0.34)\times10^{-2}$	$(7.90\pm2.09)\times10^{-3}$
Burgers 1D	$(6.70\pm2.95)\times10^{-3}$	$(2.24\pm1.20)\times10^{-3}$	$(6.74\pm2.91)\times10^{-3}$	$\mathbf{(4.51\pm1.36)\times10^{-4}}$

4.2 Error Reduction Across Problems and Instances

Table 1 places the correction alongside the raw PINN, a neural error-field control and continued PINN training. The original neural control fits a second 4×64 network to the same operator observations. The correction lowers mean error on all nine canonical benchmarks, with the largest reductions on smooth one-dimensional problems. Every Poisson 1D corrected run is below 10^{-8}. On the two-dimensional, transport and nonlinear benchmarks, paired geometric-mean improvements range from $2.36\times$ to $9.44\times$.

The paired comparison favors the correction over the original neural control on all nine benchmarks, with intervals excluding one on eight. Continued training is more accurate on Burgers by $4.73\times$; its $1.13\times$ advantage on convection is unresolved. Against a matched exact-nonlinear error network, the paired correction-to-control ratios are 0.345 on Burgers and 0.127 on Allen–Cahn. Appendix E gives this nonlinear-control comparison.

At comparable recorded cost in a separate cohort, a zero-mean direct kernel solve on the same sites is more accurate on wave, convection, Poisson 2D and Helmholtz 2D, by 4.00–$1330\times$; the correction is more accurate on Poisson 1D and reaction–diffusion, and heat is unresolved. Both models fit their hyperparameters independently (Appendix H). Finite-difference solves of the defect equation give lower paired point errors on the two-dimensional boxes and convection from $n = 33$ points per axis; the kernel correction is more accurate on the three smooth one-dimensional problems throughout the grid sweep (Appendix F).

The held-out families also have positive mean $\log_{10}$ error reductions, with geometric gains of 1.011×10^6 for Poisson, $34.84\times$ for Helmholtz, $27.43\times$ for Burgers and $1.728\times$ for convection. Instance-level results show substantial variation within families (Appendix G). On a star-shaped domain, the same scattered observation construction gives a paired gain over the raw PINN of $3353\times$. The paired geometric-mean error ratio is 5.91 in favor of a direct zero-mean kernel solve on the same sites; the archived neural and continued-training controls give gains of $2.28\times$ and $3.01\times$ in their original cohort. Appendix G details these studies and a three-dimensional anisotropic-diffusion feasibility case.

Appendix I compares a scaled neural error-field control and continued training at the complete correction cost and at a larger allowance, with the full timing protocol and paired results.

4.3 A Nonlinear Problem with Directly Specified Forcing

We prescribe the forcing on $\Omega = (0, 1)^2$ and solve

$$-\nabla\cdot(a(x, y)\nabla u) + 5u^3 = f(x, y), \qquad u|_{\partial\Omega} = 0. \tag{17}$$

The diffusivity is $a(x, y) = 0.02[1 + 0.5\sin(2\pi x)\sin(2\pi y)] \in [0.01, 0.03]$. The forcing combines Gaussian sources centred at $(0.30, 0.35)$ and $(0.70, 0.65)$, with amplitudes 2 and -1.5 and widths

Table 2: Paired error ratios $E_{\text{ours}}/E_{\text{alternative}}$ on the new nonlinear problem, geometric mean and 95% interval over seven runs. The correction uses T_i in both columns; alternatives receive T_i or $4T_i$. All four intervals favor the correction.

Alternative	T_i	$4T_i$
Scaled neural error field	$0.0128\ [0.0085, 0.0193]$	$0.0375\ [0.0161, 0.0874]$
Continued training	$0.0293\ [0.0179, 0.0479]$	$0.0583\ [0.0311, 0.1092]$

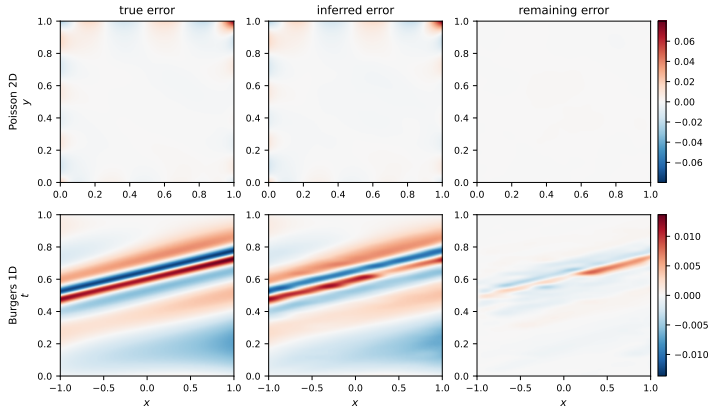


Figure 1: True error e, inferred error $\widehat{e}$ and remaining error $e-\widehat{e}$ for Poisson 2D (top) and Burgers 1D (bottom, space–time). Each row shares one colour scale; the remaining error is not rescaled. Runs follow the predefined representative-run selection rule (Appendix C).

0.12 and 0.10, respectively (Appendix J). These inputs are specified independently of a chosen solution. The numerical reference has broad positive and negative lobes; at the positive peak the cubic reaction contributes approximately 0.72 against a forcing amplitude of 2, so it accounts for a substantial part of the balance.

The frozen error operator is

$$A_\theta v = -a\Delta v - \nabla a \cdot \nabla v + 15u_\theta^2 v.$$

Diffusion, coefficient gradients and the network-dependent reaction coefficient all enter the operator observations. The same complete correction uses 500 interior and 200 boundary observations for each of seven training repetitions of this physical instance, with the fixed training budget and fitting rules of Section 4.1. An independently implemented conservative finite-volume Newton solver provides the evaluation reference on a fixed $65/129/257/513$ grid ladder. It uses midpoint face coefficients and damped Newton from the zero field; the reference is used to evaluate the resulting predictions. This tests reconstruction from prescribed equation data when the solution is available through a numerical reference.

Against the finest numerical reference, the mean relative error falls from 1.31×10^{-2} to 1.08×10^{-4}, with a paired geometric-mean improvement of $115\times$ and error-field correlation 0.99994. The complete additional correction cost averages 81.0 s. Both neural alternatives remain less accurate when allowed 324.1 s under the full-cost protocol of Appendix I (Table 2).

The final reference-grid difference is 5.17×10^{-5} relative, with observed convergence order 1.9935. A norm-based sensitivity check verifies that all 28 per-run comparisons in Table 2 retain their direction when the recorded 257-grid reference replaces the 513-grid reference; the raw-to-corrected geometric improvement stays above $71\times$ under that substitution. The reference's 513-grid CPU assembly and solve take 7.45 s, a separate classical-solver cost with different readout and validation work (Appendix J).

4.4 Error-Field Reconstruction and Component Analysis

Spatial recovery. The inferred field reproduces the boundary concentration and corner peak of the Poisson 2D error (Figure 1). For Burgers it captures the transported layer and smooth background but is displaced and split along part of the layer, where the remaining error concentrates. Operator observations recover the global field while leaving some localized structure partially resolved.

Linearization and reconstruction. A directly resolved Newton reference separates the terms in equation 14. It solves the linearized defect problem with the same frozen coefficients and boundary and initial defects as the correction. Spatial central differences and Crank–Nicolson stepping provide a grid-aligned numerical field δ_{ref}; refinement and a separate BDF2 check verify its resolution (Appendix E.1). The remaining error decomposes as

$$a = e - \delta_{\mathrm{ref}}, \qquad b = \delta_{\mathrm{ref}} - \widehat{e}, \qquad e - \widehat{e} = a + b.$$

The linearization discrepancy $\|a\|/\|e\|$ is 0.014 ± 0.009 on canonical Burgers and 0.005–0.007 on its two held-out instances. The reconstruction discrepancy $\|b\|/\|e\|$ is larger, 0.41 ± 0.30 and 0.77–0.98 respectively. Thus the resolved linearized target is close to the true error, while reconstruction from the sampled operator data accounts for the larger discrepancy.

The components' directions also matter. Writing $L = \|a\|/\|e\|$, $R = \|b\|/\|e\|$ and $T = \|a+b\|/\|e\|$, the identity is $T^2 = L^2 + R^2 + 2LR\cos\angle(a, b)$. On Allen–Cahn the components can partially cancel: in one recorded run, $L = 0.17$ and $R = 0.15$ combine to $T = 0.07$ with cosine -0.93. The decomposition therefore connects the reconstructed field to both the magnitude and spatial alignment of the error left after correction.

Prior mean and regularization. The two-by-two study crosses prior mean and stage at the inherited kernel parameters (Appendix E.3). Replacing the solution prior mean u_θ by zero at Stage 1 raises the Burgers error by $70.9\times$ ($[25.9, 194]$); its Allen–Cahn ratio is 1.05 ($[0.931, 1.17]$). The latter substitution has a larger effect at Stage 2, $34.5\times$ ($[18.8, 63.2]$), showing dependence on the regularization stage. Stage 2 itself raises the Allen–Cahn error by $3.28\times$ ($[1.06, 10.1]$), while changes on Burgers, 0.999 ($[0.978, 1.02]$), and the new nonlinear problem, 1.32 ($[0.996, 1.74]$), are unresolved. The principal comparisons use the complete two-stage procedure; Stage 1 is its component ablation.

5 Discussion and Conclusion

Observable PDE and condition defects support a continuous global correction of a trained PINN. The complete two-stage procedure recovers substantial spatial error structure across linear and nonlinear equations. On the directly forced variable-coefficient problem, error against the numerical reference falls by approximately two orders of magnitude, with lower error than both measured neural alternatives at both budgets (Section 4.3).

The comparisons distinguish error reduction from the return on additional computation. Continued training is more accurate on Burgers at matched cost, while the kernel correction retains an advantage on the new nonlinear problem even at the alternatives' larger budget. Direct kernel and finite-difference solves provide stronger alternatives on several linear problems; the correction leads on smooth one-dimensional cases (Section 4.2). Thus the PDE, starting field and available numerical structure jointly determine the accuracy tradeoff.

The Newton decomposition explains where the correction leaves error. On the studied Burgers cases, the linearized correction is close to the true error, and reconstruction from sampled operator data is the larger discrepancy. The component study further separates the reconstruction centre from regularization: Stage 2 increases Allen–Cahn error, while its effects on Burgers and the new nonlinear problem are unresolved (Section 4.4).

Dense kernel fitting sets the computational scale of this implementation. The matched studies cover smooth scalar equations in one and two spatial dimensions, supplemented by a three-dimensional feasibility case; shocks, singular solutions and larger systems remain open settings. The resulting correction is a callable field: after fitting, it can be evaluated at new query points with the network fixed.

AI USE STATEMENT

AI tools assisted with manuscript editing, and computational workflows, including experiment automation and result analysis. The authors take responsibility for the final manuscript and results.

ETHICS STATEMENT

This work concerns numerical accuracy of solvers for partial differential equations on synthetic and manufactured benchmark problems. It involves no human subjects, no personal or sensitive data, and no released dataset. We see no direct pathway to harm beyond the general risk of over-trusting a numerical correction; Section 3.4 states that the covariance does not provide calibrated coverage, and Section 5 and Appendix F identify regimes in which other methods are more accurate. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The method is fully specified by Section 3 and Algorithm 1, with the implementation-critical details — observation-vector sign, coefficient evaluation points for Gram symmetry, per-block relative regularization, and profiled amplitude — stated in Section 3 and Appendix B. All theoretical assumptions and complete proofs are collected in Appendix A. The experimental protocol, including checkpoint selection, shared starting networks, separation of reference evaluation from correction fitting, and the statistical conventions, is summarized in Section 4.1 and described in Appendix C, where per-benchmark observation-block counts are tabulated. In the held-out experiment of Section 4.2, the development/test split is fixed before any solve is evaluated, development and test physical instances are disjoint, and all runs of an instance stay on the same side of the split; the split rule and its seed are part of the supplementary material. Anonymous source code, per-run experimental records and the scripts used to generate the reported tables and figures are provided with the paper.

REFERENCES

Mark Ainsworth and J. Tinsley Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, 2000. doi: 10.1002/9781118032824.

Nivar Anwer, Marien Chenaud, José Alves, and Frédéric Magoulès. Pointwise error estimates for numerical physics-informed neural networks. *arXiv preprint arXiv:2607.03431*, 2026.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. doi: 10.1090/S0002-9947-1950-0051437-7.

Pau Batlle, Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Error analysis of kernel/GP methods for nonlinear and parametric PDEs. *Journal of Computational Physics*, 520:113488, 2025.

Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10:1–102, 2001.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `https://github.com/jax-ml/jax`.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi: 10.1137/0916069.

Lianghao Cao, Thomas O'Leary-Roseberry, Prashant K. Jha, J. Tinsley Oden, and Omar Ghattas. Residual-based error correction for neural operator accelerated infinite-dimensional Bayesian inverse problems. *Journal of Computational Physics*, 486:112104, 2023. doi: 10.1016/j.jcp.2023.112104. arXiv:2210.03008.

Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami. Probabilistic numerical methods for partial differential equations and Bayesian inverse problems. *arXiv preprint arXiv:1605.07811*, 2016. Version 3, revised 2017.

Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019. doi: 10.1137/17M1139357.

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*, 92:88, 2022. doi: 10.1007/s10915-022-01939-z.

Tim De Ryck, Ameya D. Jagtap, and Siddhartha Mishra. Error estimates for physics-informed neural networks approximating the Navier–Stokes equations. *IMA Journal of Numerical Analysis*, 44(1): 83–119, 2024. doi: 10.1093/imanum/drac085.

Peter Deuflhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, volume 35 of *Springer Series in Computational Mathematics*. Springer, 2004. ISBN 9783540210993.

Michael F. Driscoll. The reproducing kernel Hilbert space structure of the sample paths of a Gaussian process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 26:309–316, 1973. doi: 10.1007/BF00534894.

Weinan E and Bing Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6:1–12, 2018. doi: 10.1007/s40304-018-0127-z.

Zexi Fan, Yan Sun, Shihao Yang, and Yiping Lu. Physics-informed inference time scaling for solving high-dimensional partial differential equations. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=d2pUyiXwcm. arXiv:2504.16172.

Gregory E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In Alain Le Méhauté, Christophe Rabut, and Larry L. Schumaker (eds.), *Surface Fitting and Multiresolution Methods*, pp. 131–138. Vanderbilt University Press, 1997.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8c63299fb2820ef41cb05e2ff11836f5-Abstract-Datasets_and_Benchmarks_Track.html.

Philipp Hennig, Michael A. Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A*, 471:20150142, 2015. doi: 10.1098/rspa.2015.0142.

Xinquan Huang and Paris Perdikaris. PhysicsCorrect: A training-free approach for stable neural PDE simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22057–22065, 2026. doi: 10.1609/aaai.v40i26.39360. URL https://ojs.aaai.org/index.php/AAAI/article/view/39360.

Ismail Huseynov, Arzu Ahmadova, and Agamirza Bashirov. Reliable error estimation for PINNs: Lower and upper a posteriori bounds. *arXiv preprint arXiv:2606.12050*, 2026.

Ameya D. Jagtap and George Em Karniadakis. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041, 2020. doi: 10.4208/cicp.OA-2020-0164.

Ameya D. Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020. doi: 10.1016/j.cma.2020.113028.

Prashant K. Jha. Residual-based error corrector operator to enhance accuracy and reliability of neural operator surrogates of nonlinear variational boundary-value problems. *Computer Methods in Applied Mechanics and Engineering*, 419:116595, 2024. doi: 10.1016/j.cma.2023.116595. URL https://www.sciencedirect.com/science/article/abs/pii/S0045782523007193.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

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.

C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*, volume 16 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1995. doi: 10.1137/1.9781611970944.

Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464, 2001.

Ehsan Kharazmi, Zhongqiang Zhang, and George E. M. Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021. doi: 10.1016/j.cma.2020.113547.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Aleksander Krasowski, René P. Klausen, Aycan Celik, Sebastian Lapuschkin, Wojciech Samek, and Jonas Naujoks. Building trust in PINNs: Error estimation through finite difference methods. *arXiv preprint arXiv:2603.15526*, 2026.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/df438e5206f31600e6ae4af72f2725f1-Abstract.html.

Isaac Elias Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998. doi: 10.1109/72.712178.

Gregory Kang Ruey Lau, Apivich Hemachandra, See-Kiong Ng, and Bryan Kian Hsiang Low. PINNACLE: PINN adaptive collocation and experimental points selection. In *International Conference on Learning Representations*, 2024. arXiv:2404.07662.

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.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506. URL https://authors.library.caltech.edu/records/06m4r-8jf10.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL https://www.nature.com/articles/s42256-021-00302-5.

Levi D. McClenny and Ulisses M. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023. doi: 10.1016/j.jcp.2022.111722.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating PDEs. *IMA Journal of Numerical Analysis*, 43(1):1–43, 2023. doi: 10.1093/imanum/drab093.

Houman Owhadi. Bayesian numerical homogenization. *Multiscale Modeling & Simulation*, 13(3): 812–828, 2015. doi: 10.1137/140974596.

Houman Owhadi. Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games. *SIAM Review*, 59(1):99–149, 2017. doi: 10.1137/15M1013894.

Marvin Pförtner, Ingo Steinwart, Philipp Hennig, and Jonathan Wenger. Physics-informed Gaussian process regression generalizes linear PDE solvers. *arXiv preprint arXiv:2212.12474v6*, 2024.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40 (1):A172–A198, 2018.

Maziar Raissi, Paris Perdikaris, and George 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.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL https://gaussianprocess.org/gpml/.

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42159–42191, 2024. URL https://proceedings.mlr.press/v235/rathore24a.html.

Robert Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, 1995. doi: 10.1007/BF02432002.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pp. 416–426. Springer, 2001. doi: 10.1007/3-540-44581-1_27.

Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018. doi: 10.1016/j.jcp.2018.08.029.

I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967. doi: 10.1016/0041-5553(67)90144-9.

Jiuyun Sun and Yong Zhang. Physics-informed error field learning: A post-training optimization framework for physics-informed neural networks. *arXiv preprint arXiv:2608.24970*, 2026.

Kianoosh Taghikhani, Yusuke Yamazaki, Jerry Paul Varghese, Markus Apel, Reza Najian Asl, and Shahed Rezaei. Neural-initialized newton: Accelerating nonlinear finite elements via operator learning. *International Journal for Numerical Methods in Engineering*, 127(14):e70380, 2026. doi: 10.1002/nme.70380. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.70380.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0a9747136d411fb83f0cf81820d44afb-Abstract-Datasets_and_Benchmarks.html.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2.

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.

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, Hanwen Wang, and Paris Perdikaris. An expert's guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*, 2023. URL https://arxiv.org/abs/2308.08468.

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. doi: 10.1016/j.cma.2024.116813.

Yongji Wang and Ching-Yao Lai. Multi-stage neural networks: Function approximator of machine precision. *Journal of Computational Physics*, 504:112865, 2024.

Holger Wendland. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2005. doi: 10.1017/CBO9780511617539.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

Liu Yang, Xuhui Meng, and George E. Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

A Formal Analysis

This appendix collects the standing assumptions and the proofs of the results stated in Section 3, together with the regularized power-function analysis of the covariance returned by the factorization.

A.1 Standing assumptions

The kernel analysis takes place in the RKHS $\mathcal{H}_k$. Its assumptions, with dependencies specified below, are:

(A1) **Kernel.** $k : \Omega \times \Omega \to \mathbb{R}$ is symmetric and strictly positive definite, with RKHS $\mathcal{H}_k$.

(A2) **Bounded observations.** Every $\lambda_i \in \Lambda$ is bounded on $\mathcal{H}_k$, which the sufficient mixed-derivative condition of Appendix B.4 guarantees. By the Riesz representation theorem each λ_i then has a representer $\phi_i \in \mathcal{H}_k$ with $[K_{\Lambda\Lambda}]_{ij} = \langle \phi_i, \phi_j \rangle_{\mathcal{H}_k} = \lambda_i \lambda'_j k$.

(A3) **Regularization.** $\Sigma \succ 0$, hence $M = K_{\Lambda\Lambda} + \Sigma \succ 0$ and equation 12 is well defined with no rank condition on $K_{\Lambda\Lambda}$.

(A4) **Well-specification.** $e \in \mathcal{H}_k$.

Assumptions (A1)–(A3) are properties of the construction. (A4) is a deterministic modelling assumption about the unknown error, needed for the native-space error bounds. It is distinct from the auxiliary Gaussian-process reading of Section 3.3: the conditional mean of $\mathcal{GP}(0, k)$ given observations with covariance Σ coincides with $\widehat{e}$, but sample paths of $\mathcal{GP}(0, k)$ generally do not lie in $\mathcal{H}_k$ (Driscoll, 1973; Kanagawa et al., 2018). The Gaussian-process notation motivates the conditional mean and covariance formulas without establishing (A4).

Setting for the nonlinear analysis. Proposition 2 is stated in abstract spaces. Let X be a Banach space of functions on $\overline{\Omega}$ that embeds continuously in $C^2(\overline{\Omega})$, so that the interior operator and every boundary and initial-condition operator can be evaluated pointwise. Let Y be a Banach space with $f \in Y$ and $N : U \to Y$ for an open set $U \subset X$, and let Z_j be Banach spaces with $g_j \in Z_j$ and $B_j : X \to Z_j$ bounded and linear, as are the trace, normal-derivative and initial-velocity operators used here. The linearized boundary-value operator collects the linearized PDE and all boundary and initial conditions,

$$\mathcal{A}_\theta v := \big(A_\theta v,\ B_1 v,\ \ldots,\ B_J v\big) \in \mathcal{Z} := Y \times Z_1 \times \cdots \times Z_J, \tag{18}$$

with $\|(z_0, z_1, \ldots, z_J)\|_{\mathcal{Z}} = \|z_0\|_Y + \sum_j \|z_j\|_{Z_j}$. Not every element of $\mathcal{Z}$ can be the data of a solution in X: where faces carrying different conditions meet, such as the initial line and a lateral boundary, the traces of a function in X satisfy compatibility conditions. Let $\mathcal{Z}_{\mathrm{adm}} \subseteq \mathcal{Z}$ denote the admissible data space of the linearized boundary-value problem, the linear subspace of PDE and trace data that satisfy the compatibility conditions induced by the trace operators, with the norm of $\mathcal{Z}$; it contains $\mathcal{A}_\theta(X)$. The Newton correction $\delta \in X$ is the solution of the complete linearized defect problem

$$\mathcal{A}_\theta\, \delta = -\big(r,\ r_{\partial,1},\ \ldots,\ r_{\partial,J}\big), \qquad \text{that is,} \qquad A_\theta \delta = -r, \quad B_j \delta = -r_{\partial,j}\ \ (j = 1, \ldots, J). \tag{19}$$

Evaluating equation 19 at the observation sites gives $\Lambda\delta = y$ exactly. To pass from Y to the finite data, let $S : Y \to \mathbb{R}^m$ sample a PDE-data function at the m interior observation sites, $(Sq)_i = q(x_i)$, so that the sampled remainder of equation 7 is $\boldsymbol{\eta}_\Lambda = [\,-S\,R(e);\, 0;\, \ldots;\, 0\,]$, with R the Taylor remainder of Appendix A.4. Proposition 2 uses (N1)–(N4); (N5) is used only for the sampled remainder in Appendix A.6.

(N1) **Differentiability.** U is convex and contains u_θ and u, and N is Fréchet differentiable on U.

(N2) **Lipschitz derivative.** $\|N'(v) - N'(w)\|_{X \to Y} \le L_N \|v - w\|_X$ for all $v, w \in U$.

(N3) **Bounded invertibility on admissible data.** $\mathcal{A}_\theta : X \to \mathcal{Z}_{\mathrm{adm}}$ is bijective with $\|\mathcal{A}_\theta^{-1}\|_{\mathcal{Z}_{\mathrm{adm}} \to X} \le C_A$, and the defect data $(r, r_{\partial,1}, \ldots, r_{\partial,J})$ lie in $\mathcal{Z}_{\mathrm{adm}}$.

(N4) **Reconstruction in X.** The representers $\phi_1, \ldots, \phi_n$ belong to X, so $\widehat{e} \in X$; this holds, for example, when $\mathcal{H}_k$ embeds continuously in X. The condition is needed only to measure $\widehat{e}$ in the norm of X, not for the coefficient solve, and it is distinct from the well-specification assumption (A4).

(N5) **Bounded sampling.** $\|Sq\|_2 \le C_S\|q\|_Y$ for all $q \in Y$; this holds, for example, when Y embeds continuously in $C(\overline{\Omega})$.

Proposition 1 and the regularized power-function identity of Proposition 3 need only (A1)–(A3): they are statements about the regularized estimator in $\mathcal{H}_k$ and do not require the error to lie in that space. Assumption (A4), $e \in \mathcal{H}_k$, is needed only for the native-space error bounds, and it is not established for the squared-exponential kernel and tanh networks used in the experiments reported here. Proposition 2 needs (N1)–(N4), with (N5) entering only the sampled remainder of Appendix A.6. The paper does not establish a benchmark-specific choice of spaces satisfying (N1)–(N3), including the admissible trace compatibility conditions and bounded inverse in (N3). Appendix A.5 describes these requirements for the canonical nonlinear benchmarks.

The variational characterization and kernel identities hold for each kernel and positive-definite regularization matrix satisfying their assumptions. Holding the fitted quantities fixed therefore characterizes the final solve also when its parameters are selected from y. In the zero-regularization limit of Proposition 1, the kernel and D stay fixed as ε decreases.

A.2 Exact linear defect equation

Theorem 1 (Exact linear defect equation)**.** *Let L and B be linear, let u satisfy $Lu = f$ and $Bu = g$, and let u_θ be any fixed function for which the required operator evaluations exist. Then $e = u - u_\theta$ satisfies*

$$Le = f - Lu_\theta = -r, \qquad Be = g - Bu_\theta = -r_\partial. \tag{20}$$

Proof. Substitute $e = u - u_\theta$ and use linearity of L and B. □

The observed right-hand side is therefore the exact right-hand side of the error equation, independent of the quality of u_θ, so in the linear case Σ is regularization and not a model of measurement noise (Section 3.5). The analysis distinguishes three statements:

1. exactness of the *observation equation*, Theorem 1, which is algebraic;
2. the characterization of the computed estimator 12 as the solution of a *regularized problem*, Proposition 1, which holds for every $\Sigma \succ 0$; with $\Sigma \succ 0$ the estimator does not interpolate the observations;
3. adequacy of the kernel as a *model of real PINN error fields*, assumption (A4), which is not established for the experimental errors here.

The first two are established here; the third is an assumption.

A.3 Proof of Proposition 1

The argument uses the representer-span reduction underlying kernel regularization (Schölkopf et al., 2001). By (A2), $\lambda_i(v) = \langle v, \phi_i\rangle_{\mathcal{H}_k}$. Let $\Lambda^* : \mathbb{R}^n \to \mathcal{H}_k$, $\Lambda^* a = \sum_i a_i\phi_i$, be the adjoint of Λ, so that $\Lambda\Lambda^* = K_{\Lambda\Lambda}$ and $\widehat{e} = \Lambda^* c$ with $c = M^{-1}y$. No invertibility of $K_{\Lambda\Lambda}$ is used.

Reduction to the representer span. Let $\mathcal{S} = \operatorname{span}\{\phi_i\}$ and decompose $v = v_\parallel + v_\perp$ with $v_\parallel \in \mathcal{S}$ and $v_\perp \perp \mathcal{S}$. Then $\Lambda v_\perp = 0$, so $\Lambda v = \Lambda v_\parallel$, while $\|v\|^2 = \|v_\parallel\|^2 + \|v_\perp\|^2$. Hence $\mathcal{J}(v) = \mathcal{J}(v_\parallel) + \|v_\perp\|^2$, and the orthogonal component only increases the objective.

Normal equation. $\mathcal{J}$ is Fréchet differentiable on $\mathcal{H}_k$ with gradient $\nabla\mathcal{J}(v) = 2v + 2\Lambda^*\Sigma^{-1}(\Lambda v - y)$. For $v = \widehat{e}$ we have $\Lambda\widehat{e} = K_{\Lambda\Lambda}c$, and $(K_{\Lambda\Lambda} + \Sigma)c = y$ gives $\Sigma^{-1}(y - K_{\Lambda\Lambda}c) = c$, so

$$\nabla\mathcal{J}(\widehat{e}) = 2\Lambda^* c - 2\Lambda^*\Sigma^{-1}\big(y - K_{\Lambda\Lambda}c\big) = 2\Lambda^* c - 2\Lambda^* c = 0. \tag{21}$$

Uniqueness. $v \mapsto \|v\|^2_{\mathcal{H}_k}$ is strictly convex and $v \mapsto \|\Lambda v - y\|^2_{\Sigma^{-1}}$ is convex, so $\mathcal{J}$ is strictly convex. A stationary point of a differentiable convex functional is a global minimizer, and strict convexity makes the minimizer unique; hence $\widehat{e}$ is the unique minimizer in $\mathcal{H}_k$. When $K_{\Lambda\Lambda}$ is singular the ϕ_i are linearly dependent and other coefficient vectors represent the same function, but the function is unique and $c = M^{-1}y$ is well defined by (A3).

Zero-regularization limit. Let $\Sigma = \varepsilon D$ with fixed $D \succ 0$. Substituting $\widetilde{\Lambda} = D^{-1/2}\Lambda$ and $\widetilde{y} = D^{-1/2}y$ gives $\widehat{e}_\varepsilon = \widetilde{\Lambda}^*(\widetilde{\Lambda}\widetilde{\Lambda}^* + \varepsilon I)^{-1}\widetilde{y}$, the Tikhonov-regularized solution of $\widetilde{\Lambda}v = \widetilde{y}$. Because $\widetilde{\Lambda}$ has finite rank, $\widehat{e}_\varepsilon$ converges in $\mathcal{H}_k$ as $\varepsilon \to 0$ to the minimum-norm minimizer of $\|\Lambda v - y\|_{D^{-1}}$. Under the compatibility condition $y \in \operatorname{ran}\Lambda$ this minimizer satisfies $\Lambda v = y$ and is the minimum-RKHS-norm solution of the observation equations, independent of D. Compatibility holds whenever $K_{\Lambda\Lambda}$ is nonsingular, since then $\operatorname{ran}\Lambda = \mathbb{R}^n$, and for a linear PDE under (A4), since $y = \Lambda e$ by Theorem 1. Without it the limit does not match the observations. □

A.4 Proof of Proposition 2

The local remainder argument follows classical Newton analysis (Deuflhard, 2004; Kelley, 1995), with the complete boundary-value operator included explicitly. Let $R(e) = N(u_\theta + e) - N(u_\theta) - A_\theta e$, so that $\eta = -R(e)$ in equation 2. By (N1) the segment $\{u_\theta + te : t \in [0,1]\}$ lies in U, and by (N1)–(N2)

$$R(e) = \int_0^1 \big(N'(u_\theta + te) - N'(u_\theta)\big)e\,dt, \qquad \|R(e)\|_Y \le \int_0^1 L_N\, t\,\|e\|_X^2\,dt = \tfrac{L_N}{2}\|e\|_X^2. \quad (22)$$

Since $N(u) = f$, $A_\theta e = -r - R(e)$, and linearity of B_j with $B_j u = g_j$ gives $B_j e = -r_{\partial,j}$. Subtracting equation 19,

$$\mathcal{A}_\theta(e - \delta) = \big(-R(e),\, 0,\, \ldots,\, 0\big), \quad (23)$$

and this tuple, being the image of $e - \delta \in X$, lies in $\mathcal{Z}_{\mathrm{adm}}$. Hence (N3) gives $\|e - \delta\|_X \le C_A\|R(e)\|_Y \le \frac{C_A L_N}{2}\|e\|_X^2$. By (N4), $\widehat{e} \in X$, and the decomposition equation 15 follows from the triangle inequality in X applied to $e - \widehat{e} = (e - \delta) + (\delta - \widehat{e})$. □

Under (N5) the same estimate controls the sampled remainder, $\|\boldsymbol{\eta}_\Lambda\|_2 = \|S\,R(e)\|_2 \le C_S\|R(e)\|_Y \le \frac{C_S L_N}{2}\|e\|_X^2$.

The boundary and initial-condition components of equation 19 are needed: the interior equation $A_\theta\delta = -r$ alone does not determine δ, and (N3) applies only to the complete operator $\mathcal{A}_\theta$.

A.5 The nonlinear benchmarks

For viscous Burgers, $N(u) = u_t + uu_x - \nu u_{xx}$ is quadratic, so the remainder in Proposition 2 is exactly

$$R(e) = e\,e_x, \qquad \eta = -e\,e_x, \quad (24)$$

and $N'(v) - N'(w)$ acts as $h \mapsto (v - w)h_x + (v - w)_x h$. Condition (N2) therefore holds on all of X, with $L_N = 2C_{\mathrm{b}}$, whenever the bilinear map $(v, w) \mapsto v\,w_x$ is bounded from $X \times X$ to Y with constant C_{b}. The remainder contains a derivative of the error, so this bound cannot be obtained with X and Y both equal to L^2, where vw_x need not be square integrable. Condition (N3) is a well-posedness property of the linearized initial–boundary-value problem for $A_\theta v = v_t + u_\theta v_x + (u_\theta)_x v - \nu v_{xx}$ with its initial and boundary operators. Whether it holds, and with which constant, depends on ν, on u_θ and on the choice of X, Y and $\mathcal{Z}_{\mathrm{adm}}$, which must suit the parabolic structure of A_θ, the compatibility of its initial and boundary data, and the bilinear bound above. These are the requirements on a choice of spaces needed to specialize the conditional proposition to Burgers.

The Allen–Cahn benchmark has $N(u) = u_t - 10^{-4}u_{xx} + 5u^3 - 5u$ (Appendix C.1), so $A_\theta v = v_t - 10^{-4}v_{xx} + (15u_\theta^2 - 5)v$ and the remainder is exactly $R(e) = 5(3u_\theta e^2 + e^3)$, with $\eta = -R(e)$. It contains no derivative of the error, and at each point it depends only on u_θ and e there. Condition (N2) is then a local condition, with L_N depending on bounds for u_θ and u, which is why (N1)–(N2) are stated on a neighbourhood U; (N3) is again assumed.

A.6 The regularized operator power function

Probabilistic numerical methods represent uncertainty left by finite computational information (Hennig et al., 2015; Cockayne et al., 2019). Here its deterministic counterpart is a regularized power function under the stated RKHS assumptions. The conditional covariance of the auxiliary Gaussian model, returned by the same factorization as equation 12, is

$$\widehat{C}(x, x') = k(x, x') - k_\Lambda(x)^\top M^{-1} k_\Lambda(x'). \quad (25)$$

For a bounded functional F on $\mathcal{H}_k$ define the *regularized power function*

$$P^2_{F,\Sigma} \;=\; FF'k \;-\; k_{F\Lambda}M^{-1}k_{\Lambda F}, \qquad M = K_{\Lambda\Lambda} + \Sigma, \tag{26}$$

which is exactly the covariance functional the implementation evaluates; for point evaluation $F = F_x$ it equals $\widehat{C}(x,x)$.

Proposition 3 (Regularized power-function identity and bound)**.** *Under (A1)–(A3),*

$$P_{F,\Sigma} \;=\; \sup\Big\{\, |F(v)| \;:\; v \in \mathcal{H}_k,\; a \in \mathbb{R}^n,\; \|v\|^2_{\mathcal{H}_k} + a^\top\Sigma^{-1}a \le 1,\; \Lambda v + a = 0 \,\Big\}, \tag{27}$$

and, if additionally (A4) holds and $\Lambda e = y + \eta$ for some $\eta \in \mathbb{R}^n$, as in equation 7, the estimator $\widehat{e} = k_\Lambda(\cdot)^\top M^{-1} y$ satisfies

$$\big|F(e) - F(\widehat{e})\big| \;\le\; P_{F,\Sigma}\,\|e\|_{\mathcal{H}_k} \;+\; \sqrt{FF'k - P^2_{F,\Sigma}}\;\|\eta\|_{M^{-1}}, \qquad \|\eta\|_{M^{-1}} \le \|\eta\|_{\Sigma^{-1}}. \tag{28}$$

The augmented space. Under (A1)–(A3) set $\mathcal{G} = \mathcal{H}_k \oplus \mathbb{R}^n$ with $\langle (v,a),(w,b)\rangle_{\mathcal{G}} = \langle v, w\rangle_{\mathcal{H}_k} + a^\top\Sigma^{-1}b$, an inner product because $\Sigma \succ 0$. Define $\widetilde{\lambda}_i(v,a) = \lambda_i(v) + a_i$ and $\widetilde{F}(v,a) = F(v)$, both bounded on $\mathcal{G}$.

Representers and Gram matrix. The representer of $\widetilde{\lambda}_i$ is $\widetilde{\phi}_i = (\phi_i, \Sigma \mathrm{e}_i)$, since $\langle (v,a),(\phi_i,\Sigma\mathrm{e}_i)\rangle_{\mathcal{G}} = \lambda_i(v) + a^\top\Sigma^{-1}\Sigma\mathrm{e}_i = \lambda_i(v) + a_i$. Their Gram matrix is exactly M:

$$\langle \widetilde{\phi}_i, \widetilde{\phi}_j\rangle_{\mathcal{G}} = \langle \phi_i, \phi_j\rangle_{\mathcal{H}_k} + \mathrm{e}_i^\top\Sigma\Sigma^{-1}\Sigma\mathrm{e}_j = [K_{\Lambda\Lambda}]_{ij} + \Sigma_{ij} = M_{ij}. \tag{29}$$

$M \succ 0$ by (A3), so $\widetilde{\mathcal{S}} = \operatorname{span}\{\widetilde{\phi}_i\}$ has dimension n and its orthogonal projection $\widetilde{\Pi}$ is well defined. The representer of $\widetilde{F}$ is $(\phi_F, 0)$ and $\langle \widetilde{\phi}_i, (\phi_F,0)\rangle_{\mathcal{G}} = \lambda_i(\phi_F) = [k_{\Lambda F}]_i$. The regularized problem is therefore an interpolation problem in the augmented space, so the bound below applies to the estimator actually computed.

Identity. For a bounded functional on a Hilbert space, the norm of the component of its representer orthogonal to $\widetilde{\mathcal{S}}$ equals the supremum of the functional over the unit ball of the annihilator of $\widetilde{\Lambda}$; this uses only Riesz representation, orthogonal projection and Cauchy–Schwarz, never the reproducing property. Applying it in $\mathcal{G}$,

$$\|(I - \widetilde{\Pi})(\phi_F, 0)\|^2_{\mathcal{G}} = FF'k - k_{F\Lambda}M^{-1}k_{\Lambda F} = P^2_{F,\Sigma}, \tag{30}$$

and in coordinates the supremum characterization is the stated variational form 27.

Bound without observation discrepancy. Let $\eta = 0$, so $y = \Lambda e$. The projection of $(e,0)$ onto $\widetilde{\mathcal{S}}$ has coefficients solving $Mc = (\langle (e,0), \widetilde{\phi}_i\rangle)_i = (\lambda_i(e))_i = y$, hence $c = M^{-1}y$ and

$$\widetilde{\Pi}(e,0) = \Big(\textstyle\sum_i c_i\phi_i,\; \Sigma c\Big), \qquad \widetilde{F}\big(\widetilde{\Pi}(e,0)\big) = k_{F\Lambda}M^{-1}y = F(\widehat{e}). \tag{31}$$

The $\mathcal{H}_k$-component of the augmented interpolant is therefore the regularized estimator. Since $(e,0) - \widetilde{\Pi}(e,0)$ is annihilated by every $\widetilde{\lambda}_i$,

$$|F(e) - F(\widehat{e})| \le P_{F,\Sigma}\|(e,0) - \widetilde{\Pi}(e,0)\|_{\mathcal{G}} \le P_{F,\Sigma}\|(e,0)\|_{\mathcal{G}} = P_{F,\Sigma}\|e\|_{\mathcal{H}_k}, \tag{32}$$

the second inequality because $\widetilde{\Pi}$ is an orthogonal projection.

Bound with observation discrepancy. In general $\Lambda e = y + \eta$ and, by linearity of $y \mapsto \widehat{e}$, $F(\widehat{e}_{\Lambda e}) - F(\widehat{e}_y) = k_{F\Lambda}M^{-1}\eta$. Cauchy–Schwarz in the M^{-1} inner product gives $|k_{F\Lambda}M^{-1}\eta| \le (k_{F\Lambda}M^{-1}k_{\Lambda F})^{1/2}(\eta^\top M^{-1}\eta)^{1/2} = \sqrt{FF'k - P^2_{F,\Sigma}}\,\|\eta\|_{M^{-1}}$, and $M \succeq \Sigma$ gives $\|\eta\|_{M^{-1}} \le \|\eta\|_{\Sigma^{-1}}$. Combining with the previous paragraph by the triangle inequality proves the proposition. □

Regularization enlarges the admissible set. The constraint in equation 27 is $\Lambda v + a = 0$, not $\Lambda v = 0$, so the regularized power function measures unresolved directions up to a Σ-weighted tolerance in the observations. Whenever $K_{\Lambda\Lambda}$ is also nonsingular, $M \succeq K_{\Lambda\Lambda}$ gives $M^{-1} \preceq K^{-1}_{\Lambda\Lambda}$ and hence $P_{F,\Sigma} \ge P_F$: the regularized quantity is the larger, more conservative one.

The discrepancy term vanishes in the linear case. The second term of equation 28 is the only route by which observation discrepancy enters. For a linear PDE, Theorem 1 gives $\eta = 0$ identically, so the bound reduces to $P_{F,\Sigma}\|e\|_{\mathcal{H}_k}$ with no residual approximation. For a nonlinear PDE the discrepancy is $\boldsymbol{\eta}_\Lambda = [-S\,R(e);\,0;\,\ldots;\,0]$, and under (N1), (N2) and (N5) $\|\boldsymbol{\eta}_\Lambda\|_2 \le \frac{C_S L_N}{2}\|e\|_X^2$ (Appendix A.4), so it is second order in the error. The bound measures it in the Σ^{-1} metric set by the fitted regularization level, with $\|\boldsymbol{\eta}_\Lambda\|_{\Sigma^{-1}} \le \lambda_{\min}(\Sigma)^{-1/2}\|\boldsymbol{\eta}_\Lambda\|_2$. In the linear case the covariance therefore bounds the error of the computed estimator through the first term alone; in the nonlinear case the bound also depends on how well Σ absorbs the remainder.

A.7 Integrated bound and scope

Taking $F = F_x$, squaring and integrating the pointwise bound against a probability measure μ gives, in the linear case,

$$\|e - \widehat{e}\|_{L^2(\mu)} \;\le\; \|e\|_{\mathcal{H}_k}\Big(\int_\Omega \widehat{C}(x,x)\,d\mu(x)\Big)^{1/2}. \tag{33}$$

The analysis establishes three properties of the observation equation and the kernel estimator:

1. for linear PDEs the defect observations are exact, however inaccurate u_θ is;
2. conditional on its hyperparameters the correction is a globally solved operator-kernel estimator;
3. the covariance measures unresolved directions of the assumed RKHS under the regularized observations and, in the linear case under (A4), controls the error of the computed estimator.

The right-hand side of equation 33 contains the unknown factor $\|e\|_{\mathcal{H}_k}$, and (A4) is not established for the experimental error fields, so $\widehat{C}$ does not provide calibrated coverage for real PINN error fields.

B Operator and Implementation Details

B.1 Automatic extraction of the linearized operator

For a nonlinear N we recover the coefficient fields of

$$A_\theta v = M(x) : \nabla^2 v + b(x)\cdot\nabla v + c(x)\,v \tag{34}$$

pointwise, by exact Jacobian–vector products of N against the local Taylor basis $\{1,\ x_i - p_i,\ (x_i - p_i)(x_j - p_j)\}$ at each site p. The same procedure applies to every benchmark, so no operator-specific derivatives are coded by hand. Because the contraction $M : \nabla^2 v$ sums over all index pairs, the symmetric entries M_{ij} and M_{ji} each contribute to a quadratic probe; the probe values are therefore halved and the recovered matrix is symmetrized.

B.2 Variable coefficients and kernel arguments

Applying A_θ to a kernel argument reduces to forward-mode differentiation of the kernel. Interior Gram entries compose the operator on both arguments, $A_{\theta,x}A_{\theta,x'}k$, with each coefficient field evaluated at its own point: M, b, c taken at x in the outer application and at x' in the inner one. This preserves the symmetry of the Gram matrix for spatially varying coefficients, as in the nonlinear case; evaluating both factors at the same point does not. Assembled variable-coefficient Gram matrices are checked for exact symmetry and positive definiteness.

B.3 Observation blocks and boundary operators

A block's operator is the one the physical condition imposes: a Dirichlet face contributes a trace, a Neumann face a normal derivative, an initial-time face either point evaluation or ∂_t for a prescribed initial velocity. The observed vector carries the sign of equation 6: in the linear case $y = \Lambda e$ exactly, and the opposite sign would negate the correction. The observation and kernel assembly require no mesh. All reported experiments use $m = 500$ interior entries. On the canonical benchmarks the interior operator sites are placed using Sobol sequences (Sobol', 1967) with scrambling seeds derived

deterministically from the run seed and benchmark name. The boundary and initial-condition sites are 200 uniform random samples in the bounding box, allocated near-equally across the condition groups and projected onto the face or initial line of their group. Coordinates that coincide after projection are compressed to one unique entry whose multiplicity is kept as an explicit block weight: on Poisson 1D the 100 samples of each face all map to the endpoint, so each endpoint block is one trace entry of weight 100, giving 502 unique entries, and the other canonical benchmarks have 700 (Table 6). These correction sites are drawn independently of the collocation points used to train the network. The irregular domain and the three-dimensional case use the placements of Appendix G, and the latter has 800 observations.

As a worked instance, take $-u'' = f$ on $(0,1)$ with $u(0) = g_0$ and $u(1) = g_1$, so that $r = -u_\theta'' - f$ and the error relations are $-e''(x_i) = -r(x_i)$, $e(0) = g_0 - u_\theta(0)$ and $e(1) = g_1 - u_\theta(1)$. Each interior functional is $\lambda_i(v) = -v''(x_i)$, so its representer $\phi_i(x) = -\partial_{x'}^2 k(x,x')|_{x'=x_i}$ is a second derivative of the kernel centred at a site, and the two Dirichlet observations contribute the slices $k(x,0)$ and $k(x,1)$. An interior–interior Gram entry applies the second-order operator to both kernel arguments, and the two factors of -1 cancel, leaving $[K_{\Lambda\Lambda}]_{ij} = \partial_x^2 \partial_{x'}^2 k(x,x')|_{x=x_i,x'=x_j}$; an interior–boundary entry applies the operator to one argument and a trace to the other. Solving equation 11 then returns a correction assembled from these functions, following the chain $r(x_i) \to \lambda_i \to \phi_i \to K_{\Lambda\Lambda} \to \widehat{e}$.

B.4 Kernel differentiability

The requirement on k is that every functional in Λ be bounded on $\mathcal{H}_k$. For a linear differential functional of order μ evaluated at a point, a sufficient condition is that the mixed partial derivatives $\partial_x^\alpha \partial_{x'}^\beta k$ exist and are continuous on $\Omega \times \Omega$ for all $|\alpha|, |\beta| \le \mu$. The three block types occurring in this paper give the requirement directly: the interior block applies a second-order operator on both arguments, so $\mu = 2$ and mixed derivatives of total order four are needed; a Dirichlet trace block has $\mu = 0$ and needs none; the prescribed initial-velocity block on the wave equation has $\mu = 1$ and needs total order two. Mixed interior–boundary entries take the larger order on each argument, so total order four is the requirement for the whole Gram matrix and no higher derivative is formed. For a Matérn kernel of smoothness ν the condition holds when $\nu > \mu$, so $\nu = 5/2$ suffices for a second-order operator; the squared exponential satisfies it for every μ.

B.5 Block-scaled regularization

Kernel interpolation couples approximation accuracy and numerical conditioning (Schaback, 1995); the block scaling below addresses the different operator scales in this system. Applying a second-order operator to both kernel arguments introduces inverse fourth powers of the lengthscale that a Dirichlet trace block does not have. In our benchmarks the interior and boundary blocks therefore differ in mean prior variance by up to five orders of magnitude, and an unscaled diagonal regularizes them unequally: whatever value stabilizes the interior block over-regularizes the boundary data, or the reverse. Adaptive nuggets use block-trace scaling in Chen et al. (2021, Appendix A.1) and individual Gram-diagonal entries in Batlle et al. (2025, Appendix C); here we use block-mean prior variances with explicit multiplicity weights. In equation 16, D is diagonal, and the entry of each observation is the mean of the diagonal of the amplitude-free Gram matrix $\widetilde{K}$ (with $K_{\Lambda\Lambda} = s^2\widetilde{K}$) over that observation's block, divided by the block's explicit weight. The weight is 1 for 35 of the 37 canonical blocks and 100 for the two Poisson-1D endpoint blocks, where it is the multiplicity of the compressed duplicate trace observations (Appendix B.3), so that multiplicity is retained through the relative regularization scale. D therefore does not depend on the profiled amplitude, and σ_{rel} is a single relative level that means the same thing in every block. The matrix factorized both during fitting and for the reported corrections is

$$\widetilde{M} = \tfrac{1}{2}\big(\widetilde{K} + \widetilde{K}^\top\big) + \big(\sigma_{\mathrm{rel}}^2 + \tau\big)D, \qquad M = s^2\widetilde{M}, \tag{35}$$

with τ the numerical jitter of Appendix B.6. The jitter is thus part of the regularization used for prediction and is block-scaled on the same footing as σ_{rel}, rather than a separate stabilization of the factorization; Propositions 1 and 3 apply with this effective Σ.

The meaning of Σ differs by regime. For a linear PDE, Theorem 1 makes $y = \Lambda e$ exact, so Σ carries no observation error and acts only as regularization and numerical tolerance; the estimator it

defines is the regularized minimum-norm estimator of Proposition 1, which does not interpolate the observations when $\Sigma \succ 0$. For a nonlinear PDE, $\Lambda e = y + \boldsymbol{\eta}_\Lambda$ as in equation 7, and the additional regularization is intended to tolerate this discrepancy in aggregate. It is not a model of independent, zero-mean errors across sites. Proposition 3 displays the discrepancy term in the conditional error bound.

B.6 Profiled hyperparameters and numerical stabilization

We fit kernel hyperparameters through the Gaussian marginal objective (Rasmussen & Williams, 2006), with the profiled amplitude and block regularization specified below. The kernel is the anisotropic squared exponential $\exp\big(-\frac{1}{2}\sum_a (x_a - x'_a)^2/\ell_a^2\big)$ with one lengthscale ℓ_a per coordinate, in the physical coordinates of the benchmark, multiplied by the scalar amplitude s^2. The amplitude is not optimized but profiled,

$$\widehat{s}^2 = \tfrac{1}{n}\, y^\top \widetilde{M}^{-1} y, \tag{36}$$

on the full observation set, with $\widetilde{M}$ from equation 35. Outside the zero-defect branch described below, the remaining parameters minimize the profiled negative log marginal likelihood $\frac{n}{2}\log \widehat{s}^2 + \frac{1}{2}\log\det \widetilde{M}$. The optimization variables are $\log \ell_a$ and $\log \sigma_{\mathrm{rel}}$, with bounds $\ell_a \in [10^{-3}, 2]$ and $\sigma_{\mathrm{rel}} \in [10^{-10}, 10]$ and initial $\sigma_{\mathrm{rel}} = 10^{-6}$. The profiled objective and its gradient are evaluated by automatic differentiation in JAX (Bradbury et al., 2018) and supplied together to SciPy's (Virtanen et al., 2020) L-BFGS-B implementation (Byrd et al., 1995), which is run for at most 150 iterations with `ftol` $= 10^{-12}$ and `gtol` $= 10^{-8}$, both tightened from the SciPy defaults, from five starts: four isotropic starts with every lengthscale equal to 0.05, 0.15, 0.3 or 0.6, and one start at the best point of a coarse grid formed by the Cartesian product of these lengthscale values over the coordinates and $\sigma_{\mathrm{rel}} \in \{10^{-8}, 10^{-5}\}$. The start with the best finite objective is kept. The jitter follows the ladder $\tau \in \{10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}\}$ and is escalated only when the objective is non-finite or the Cholesky factorization fails. The rung is recorded for every solve. Every default float64 run reported in this paper stayed on the base rung $\tau = 10^{-10}$; the float32 precision ablation of Appendix D is the one reported arm that escalates, reaching $\tau = 10^{-4}$ during the fit in all of its runs.

When the maximum absolute observation is at most 10^{-14}, the implementation takes a separate zero-defect limit: it skips marginal-objective optimization, uses the default lengthscales 0.15 and relative regularization 10^{-6}, and returns a zero correction and zero profiled amplitude. This numerical branch avoids evaluating the logarithmic objective at zero amplitude; the positive-amplitude characterization applies to the remaining cases.

On nonlinear problems the correction has two stages, used for the canonical Allen–Cahn and Burgers benchmarks and for the problem in Appendix J. Stage 1 is the same five-start L-BFGS-B fit, which selects the lengthscales and an initial relative nugget σ_{rel}. Stage 2 then enlarges the regularization with a heuristic intended to accommodate the sampled Taylor-remainder discrepancy $\boldsymbol{\eta}_\Lambda$ of equation 7, at the Stage-1 lengthscales; it performs no further hyperparameter optimization.

Write $\widehat{e}_1$ for the Stage-1 reconstruction and D_{int} for the mean of D over the interior rows. In the following rule, $\|\widehat{e}_1\|_\infty$ denotes the maximum absolute prediction on the fixed evaluation grid, using site coordinates and predictions only:

$$\sigma_{\mathrm{after}} = \sqrt{\sigma_{\mathrm{rel}}^2 + \rho^2}, \qquad \rho = \frac{C_N\, \|\widehat{e}_1\|_\infty^2}{\sqrt{\widehat{s}^2 D_{\mathrm{int}}}}, \tag{37}$$

Here $\widehat{s}^2$ is the Stage-1 profiled amplitude of equation 36. The *amplitude heuristic regularization rule* combines the probe curvature C_N with the squared correction amplitude and normalizes by the Stage-1 interior prior scale. For Burgers, the remainder $e\, e_x$ also depends on the error derivative, so a squared-amplitude rule is a regularization heuristic rather than a Taylor-remainder bound.

Stage 2 updates the common regularization level in equation 16, so every block receives the new level while retaining its scale in D. The sampled Taylor remainder enters only the interior equations; the update acts on the coupled reconstruction, including boundary and initial-condition blocks. To estimate C_N, at each of the first $\min(64, m)$ interior sites we form the second-order Taylor basis $\{1\} \cup \{x_a - p_a\} \cup \{(x_i - p_i)(x_j - p_j)\}_{i \le j}$, evaluate every mixed second derivative $\partial_\varepsilon \partial_\eta N(u_\theta +$

$\varepsilon\varphi + \eta\psi)(p)$ at $\varepsilon = \eta = 0$ by nested forward-mode differentiation, and take C_N to be the largest of $\frac{1}{2}|\partial_\varepsilon \partial_\eta N|$ over all probe pairs and all sites. For Burgers this returns $C_N = 0.5$ exactly in every run, the value implied by $N''(u)[v, w] = vw_x + wv_x$; for Allen–Cahn it ranges over 14.33–14.94.

This regularization update is applied once, with no iteration, clipping, floor or maximum beyond a numerical guard against a vanishing denominator. The lengthscales, amplitude-free kernel, block scales D, jitter and operator coefficients are held at their Stage-1 values. After replacing σ_{rel} by σ_{after} in $\widetilde{M}$, the amplitude is re-profiled by equation 36; both $K_{\Lambda\Lambda} = \widehat{s}^2\widetilde{K}$ and $\Sigma = \widehat{s}^2(\sigma^2_{\mathrm{after}} + \tau)D$ use this updated amplitude in the final solve and readout. No further optimizer is run. The final nonlinear prediction therefore uses σ_{after} in place of σ_{rel} in equation 35, after the Stage-1 L-BFGS-B fit. Proposition 2 describes the linearization error; the Stage-2 rule sets a reconstruction regularization level using the sampled correction amplitude. In practice the fitted σ_{rel} sits near its lower bound on both canonical nonlinear benchmarks, so $\sigma_{\mathrm{after}} \approx \rho$. Appendix E.3 measures the rule's effect by re-running both canonical nonlinear benchmarks with and without it: on Burgers it changes the final error by a factor of 0.999, and on Allen–Cahn it multiplies the error by 3.28, with the respective paired intervals reported there.

B.7 Precision, coincident coordinates, and implementation tests

PINN training and kernel correction run on a single NVIDIA RTX 4090 GPU. Numerical computations use float64 except the float32 precision ablation of Appendix D; classical reference solvers have the execution details reported in their respective studies. Float32 arithmetic degrades the correction without necessarily causing a factorization failure. Kernels implemented through a radial square root need care at coincident points: under vectorization both branches of a conditional may be evaluated, and the derivative of $\sqrt{q}$ at $q = 0$ is unbounded, which makes the fourth-order derivatives of the interior block non-finite. We use a double-`where` idiom that gives the inactive branch a benign argument, and test finiteness of the profiled objective and its gradient at coincident coordinates.

Implementation checks cover Gram symmetry and positive definiteness, recovery of analytic perturbations, the zero-residual limit, and agreement between the kernel-ridge and Gaussian-process formulations. Further checks compare functional corrections with direct evaluation and leave-one-out residuals with explicit refits, verify benchmark forcing and derivatives at coincident points, and confirm identical starting networks across paired arms. The full evaluation path and separation of reference solutions from fitting are also tested.

B.8 Computational details

Each canonical benchmark uses at most 700 observations (Appendix C). The canonical fit-and-readout bracket starts after observation construction and takes 7.7 ± 1.3 s on Poisson 1D to 69.9 ± 3.5 s on Burgers (Table 7). Most of this time is spent in the low-dimensional hyperparameter optimization. The later budget study includes observation construction in its complete correction cost (Appendix I).

B.9 Fitting inputs and comparator implementations

The fitting implementation uses the frozen network, the known PDE data and the observation sites. Automated dependency and interface checks keep benchmark reference solutions and training-variant labels outside the estimator. Paired arms share the cached network described in Appendix C.3; the supplementary records retain the cache keys and execution identifiers.

We implement the neural error-field and exact-nonlinear error-network controls for this study, with the architecture and optimization settings in Table 5. The error-field control fits the operator targets y; the exact-nonlinear control fits $N(u_\theta + \varphi) - f = 0$ with the observed boundary data. Both are study-specific implementations of their governing principles.

B.10 Run cohorts and timing comparisons

Each study uses a separately recorded cohort. The cohorts are: the canonical benchmarks and the two matched comparators (Tables 7 and 8), the zero-mean direct solver (Table 28), the implemen-

tation ablations (Table 14), the training recipes (Table 12), the training-budget sweep (Table 13), the exact-nonlinear control (Table 15), the finite-difference comparator (Table 20), the held-out instances (Tables 21, 24), the irregular domain (Tables 25, 26), the three-dimensional case (Table 27), and the Newton and Batlle-style studies of Appendix E.

Paired ratios and matched wall-clock comparisons use runs from the same cohort. The correction and the direct solver were timed in one execution and the canonical table in another on the same hardware, which is why the correction's recorded time differs between Table 7 and Appendix H while its errors do not. Timings are single measurements per run and describe the costs of these recorded configurations.

C Benchmark Definitions and Experimental Protocol

Cross-equation evaluations such as PDEBench (Takamoto et al., 2022) and PINNacle (Hao et al., 2024) make the task definition, geometry and evaluation measure explicit. The problems and evaluation measures used in this study are specified below.

C.1 Canonical Benchmark Definitions

Each canonical benchmark is a specified differential operator with a closed-form reference solution u (Table 3); no numerical reference solve is used. Where the right-hand side is f, the forcing is obtained by applying the specified operator to u with automatic differentiation; where it is 0, u solves the homogeneous equation. Every Dirichlet, initial-value and initial-velocity target is the corresponding trace of the same u. Time-dependent problems are initial–boundary-value problems on $t \in [0, 1]$ with no terminal-time data, and convection uses the inflow formulation, with data on the inflow face $x = 0$ and at $t = 0$ and no outflow or periodic condition. The canonical Helmholtz problem has unit wavenumber: on $[-1, 1]^2$ the smallest homogeneous-Dirichlet eigenvalue of $-\Delta$ is $\pi^2/2 > 1$, so the Dirichlet realization of $\Delta + 1$ is negative definite. The wave speed is 2 and the convection speed 30. Burgers uses $\nu = 0.01/\pi$, $C = 0.5$ and $A = 0.1$, a travelling layer of width $2\nu/A$ centred at $x = 0$ at $t = 0$ and moving with speed C.

Table 3: Canonical benchmark definitions and closed-form reference solutions u. Five benchmarks are homogeneous, with $f \equiv 0$ and a classical exact solution; on the other four the right-hand side is manufactured from u by automatic differentiation. Domains are in x for Poisson 1D, in (x, y) for the two-dimensional problems and in (x, t) for the time-dependent problems. D: Dirichlet data on the listed faces; IC: value at $t = 0$; IV: initial velocity $\partial_t u$ at $t = 0$.

Benchmark	Equation	Domain	Reference solution u	Conditions
Poisson 1D	$-u_{xx} = f$	$[0, 1]$	$\sin \pi x + 0.3 \sin 4\pi x$	D: $x = 0, 1$
React.–diff. 1D	$u_t - 0.1u_{xx} - u = 0$	$[0, 1]^2$	$e^{(1-0.1\pi^2)t} \sin \pi x$	D: $x = 0, 1$; IC
Heat 1D	$u_t - 0.1u_{xx} = 0$	$[0, 1]^2$	$e^{-0.1\pi^2 t} \sin \pi x$	D: $x = 0, 1$; IC
Wave 1D	$u_{tt} - 4u_{xx} = 0$	$[0, 1]^2$	$\sin \pi x \cos 2\pi t$	D: $x = 0, 1$; IC; IV
Poisson 2D	$-\Delta u = f$	$[0, 1]^2$	$\sin 2\pi x \sin 2\pi y + 0.3 \sin 6\pi x \sin 4\pi y$	D: four faces
Helmholtz 2D	$\Delta u + u = f$	$[-1, 1]^2$	$\sin \pi x \sin 4\pi y$	D: four faces
Convection 1D	$u_t + 30u_x = 0$	$[0, 2\pi] \times [0, 1]$	$\sin(x - 30t)$	D: $x = 0$; IC
Allen–Cahn 1D	$u_t - 10^{-4}u_{xx} + 5u^3 - 5u = f$	$[0, 1]^2$	$\sin \pi x \cos \pi t$	D: $x = 0, 1$; IC
Burgers 1D	$u_t + uu_x - \nu u_{xx} = 0$	$[-1, 1] \times [0, 1]$	$C - A \tanh\big(A(x - Ct)/(2\nu)\big)$	D: $x = \pm 1$; IC

Evaluation. On the canonical benchmarks the reported error is the relative discrete L^2 error $\|u - \widehat{u}\|_2/\|u\|_2$, an unweighted vector two-norm over the fixed evaluation grid: 1,000 uniformly spaced points on Poisson 1D and a 200×200 uniform tensor grid on the two-dimensional and space–time benchmarks, endpoints included.

The evaluation grids and measures for the remaining studies are:

Study	Norm	Evaluation set
Canonical suite	unweighted discrete L^2	1,000 points (Poisson 1D); 200 per axis otherwise
Held-out families	unweighted discrete L^2	256 points (Poisson family); 200 per axis otherwise
Three-dimensional case	unweighted discrete L^2	$24^3 = 13{,}824$ uniform tensor grid
Irregular star domain	quadrature-weighted L^2	polar-area quadrature nodes, interior only

The held-out one-dimensional family uses 256 uniformly spaced evaluation points, compared with 1,000 for canonical Poisson 1D. Its two-dimensional and space–time families retain 200 points per axis. On the star domain, polar-area quadrature restricts evaluation to the domain interior and accounts for the area element (Appendix G).

C.2 Primary PINN Training

PINN accuracy depends on architecture and training choices (Wang et al., 2023). The primary recipe is fixed as follows to provide the same base network to every post-hoc arm within a run. Each frozen network is fully connected with four hidden layers of width 64, tanh activations and a scalar linear output (12,673 parameters for one input coordinate, 12,737 for two), initialized with Glorot variance scaling (Glorot & Bengio, 2010), normal draws and zero biases. It is trained with Adam (Kingma & Ba, 2015) at a constant learning rate of 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, without weight decay or learning-rate schedule. The loss is the mean squared PDE residual at 4,000 interior collocation points plus 100 times the mean squared condition residual at 400 boundary and initial-condition points; the boundary, initial-value and initial-velocity residuals are concatenated into one residual vector and share this single weight, so the conditions are enforced softly. These training collocation points are drawn uniformly at random once per run, split near-equally across the condition groups, and held fixed; they are distinct from the correction observation sites of Appendix B.3. The frozen network is the final iterate at the calibrated budget. All runs use float64 on a single NVIDIA RTX 4090 GPU, with seven runs, seeds 0 to 6, per canonical configuration.

The training budget (Table 4) was calibrated once per benchmark on the ladder 5,000, 10,000, 20,000, 40,000 and 80,000 steps, using calibration seed 0 and taking the first rung whose calibration-run relative error fell within $[10^{-3}, 10^{-2}]$. The selected budget was then held fixed across repeated runs and all comparison arms. It fixes a training length rather than a common final accuracy, and the raw PINN error differs across benchmarks (Table 1).

Table 4: Calibrated primary-PINN training budgets, held fixed across repeated runs and arms.

Benchmarks	Training steps
Burgers 1D, heat 1D, reaction–diffusion 1D	5,000
Poisson 1D, Helmholtz 2D, wave 1D, Allen–Cahn 1D	10,000
Poisson 2D	40,000
Convection 1D	80,000

C.3 Pairing and Comparator Protocols

Within a benchmark and run, every arm wraps the same frozen network, recovered from a content-addressed cache whose key covers the benchmark, physical instance, seed, training recipe, step budget, collocation counts and geometry, and which stores the Adam moments and step counter with the weights. The key does not depend on the number of correction observations, and paired arms differ only in the correction mechanism. Table 5 specifies the three matched comparators; the operator-kernel settings are given in Appendix B.6.

Appendix I adds a fourth arm, a neural error-field control with an operator-matched output scale, a zeroed output layer and an L-BFGS-B refinement, run on one trajectory with a total budget of four times the correction's own cost and read at one and four times it. The first readout is Adam-only, since L-BFGS-B starts at twice that cost. Its full recipe, budget accounting and results are in Appendix I. This control is evaluated in the full-cost cohort; Table 1 reports the original control under the fit-and-readout budget.

Table 5: Protocols of the matched comparators. All three wrap the same frozen network as the operator-kernel correction, use its fit and readout time as wall-clock budget, and select checkpoints by observable losses. This timing bracket excludes observation construction; the full-cost cohort of Appendix I includes it.

	Matched neural error-field control	Matched exact-nonlinear error network	Matched-time continued training
Benchmarks	all nine	Burgers, Allen–Cahn	all nine
Network	new network with widths $(d, 64, 64, 64, 64, 1)$, tanh, Glorot normal per restart	new network with widths $(2, 64, 64, 64, 64, 1)$, tanh	the frozen PINN, resumed
Objective	mean squared misfit to the operator data y at the interior sites plus 100 times that at the condition sites	$N(u_\theta + \varphi) - f = 0$ at the interior sites plus 100 times the misfit to the observed boundary and initial error data	primary physics loss (Appendix C.2)
Sites	those of the operator-kernel correction	those of the operator-kernel correction	primary training collocation points
Optimizer	Adam; learning rate 10^{-3} to step 1,000, 3×10^{-4} to step 5,000, then 10^{-4}	as for the error-field control	Adam at constant 10^{-3}, resuming moments and step counter
Restarts	three, advanced round-robin in blocks of 100 steps	three	none
Budget	initialization, compilation, optimization and checkpoint evaluation charged	as for the error-field control	optimization and loss evaluations charged
Checkpoint	every 100 steps; minimum observable operator-data loss	every 100 steps; minimum observable exact nonlinear operator-data loss	every 100 steps; minimum observable physics loss in the window

Because the matched window is defined in wall-clock time rather than update count, the relative extension of the original optimization differs substantially across benchmarks: over the seven runs, continued training performs 13,900 to 15,000 additional steps on Burgers, on top of a 5,000-step base run, and 1,000 to 1,900 additional steps on convection, on top of an 80,000-step base run. Both neural controls are implemented from their governing principles (Appendix B.9).

Isolation of reference solutions. The correction implementation takes the frozen network, known PDE data and observation sites as inputs. Reference solutions are confined to evaluation code, with dependency and interface tests checking this separation (Appendix B.9). The hyperparameter fit, site selection, regularization and covariance calculations use these correction inputs.

Field visualizations. Figure 1 uses a predefined representative-run selection rule. For each benchmark shown, the $\log_{10}$ raw error, the $\log_{10}$ corrected error and the error-field correlation are standardized across the seven runs, and the run with the smallest Euclidean distance to the benchmark mean of these three outcome metrics is displayed. Observation sites are not shown, because their coordinates are not part of the stored field data.

Observation blocks. Table 6 lists every observation block with its number of unique entries and its explicit weight. Every benchmark has 500 interior operator entries, and the boundary and initial-condition entries are sampled as described in Appendix B.3. On steady Poisson 1D the 100 projected samples of each face coincide at the endpoint, so each endpoint block holds one unique trace entry with explicit weight 100, and the system has $500 + 2 = 502$ unique entries; on the other benchmarks 200 distinct boundary and initial-condition entries bring the total to 700. Of the 37 canonical observation blocks, 35 have weight 1 and the two Poisson-1D endpoint blocks have weight 100. *Trace* blocks are Dirichlet conditions, and the *coordinate derivative* block is the prescribed initial velocity of the wave equation. No canonical benchmark has a Neumann block, so the normal-derivative operator of Section 3.2 is not exercised by these experiments.

Table 6: Per-benchmark observation blocks: the operator of each block, its number of unique entries and its explicit weight.

Benchmark	Observation block	Operator	Unique entries	Weight
Poisson 1D	interior_operator	PDE operator	500	1
	x_lower_value	trace	1	100
	x_upper_value	trace	1	100
Reaction–diffusion 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	66	1
	x_lower_value	trace	67	1
	x_upper_value	trace	67	1
Heat 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	66	1
	x_lower_value	trace	67	1
	x_upper_value	trace	67	1
Wave 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	50	1
	t_initial_velocity	coordinate derivative	50	1
	x_lower_value	trace	50	1
	x_upper_value	trace	50	1
Poisson 2D	interior_operator	PDE operator	500	1
	x_lower_value	trace	50	1
	x_upper_value	trace	50	1
	y_lower_value	trace	50	1
	y_upper_value	trace	50	1
Helmholtz 2D	interior_operator	PDE operator	500	1
	x_lower_value	trace	50	1
	x_upper_value	trace	50	1
	y_lower_value	trace	50	1
	y_upper_value	trace	50	1
Convection 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	100	1
	x_inflow_value	trace	100	1
Allen–Cahn 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	66	1
	x_lower_value	trace	67	1
	x_upper_value	trace	67	1
Burgers 1D	interior_operator	PDE operator	500	1
	t_initial_value	trace	66	1
	x_lower_value	trace	67	1
	x_upper_value	trace	67	1

Reference-based budget calibration. The canonical training budget was chosen from the first ladder rung whose seed-0 relative L^2 error against the reference solution fell in the target window. All seven runs, including seed 0, then used that budget. Thus reference information enters the base-training budget design, while correction fitting uses the known f, $\{g_j\}$ and frozen network. The six-run sensitivity analysis in Appendix D omits seed 0 from the summary but retains the reference-calibrated budgets.

C.4 Statistical Reporting

Relative errors use the unweighted discrete norm of Appendix C.1 on each study's evaluation set, except on the irregular domain, which uses the quadrature-weighted norm of Appendix G. Error-field correlation is the centered Pearson correlation between $\widehat{e}$ and e on that evaluation set: equal weights on the canonical, three-dimensional and new nonlinear grids, and the normalized polar-area quadrature weights on the star domain. Each vector is centered by its corresponding weighted mean. The implementation returns zero when a centered norm vanishes (or the weighted denominator is at machine-tiny scale on the star domain). Relative errors, error-field correlations and runtimes are reported as the arithmetic mean $\pm$ sample standard deviation over independent runs. A multiplicative improvement is the geometric mean $10^{\bar{d}}$, where $\bar{d}$ is the mean over runs of the paired reduction

$\log_{10} \mathrm{err}_{\mathrm{raw}} - \log_{10} \mathrm{err}_{\mathrm{corrected}}$ on the same frozen network; arithmetic means of raw-to-corrected ratios are avoided because they are unstable when the corrected error reaches the observed error floor. Comparisons between arms use the paired geometric-mean error ratio, formed within each run of a benchmark and not averaged across benchmarks, since the benchmarks at the error floor would dominate such an average. Held-out experiments use the physical instance as the statistical unit: repeated runs are averaged within an instance before statistics are taken across instances. Cells in which an arm is not applicable, such as causal weighting on steady problems and the linear direct solver on nonlinear PDEs, are recorded as not applicable. The wave finite-difference comparator was applicable and run, but its tested discretization diverged under refinement. This numerical failure is retained in Table 20 and excluded from accuracy-ranking summaries; it is not an inapplicable arm. Paired ratios whose confidence intervals contain one are reported as unresolved at the available sample size; equivalence and noninferiority require dedicated tests.

D Additional Canonical Results

This appendix reports the full canonical results summarized in Table 1 and three robustness studies. The canonical suite, the training recipes of Table 12 and the budget sweep of Table 13 use separate cohorts. Pairing and matched timing comparisons are made within each cohort (Appendix B.10). Values are means $\pm$ sample standard deviations over seven independent runs, and improvements are geometric means of the paired $\log_{10}$ reduction against the raw PINN (Appendix C.4). On Poisson 1D every corrected run lies below 10^{-8}, and on reaction–diffusion one run does.

Canonical benchmarks and paired comparisons. Table 7 adds the improvement, error-field correlation and wall-clock of the operator-kernel correction. Table 8 gives per-benchmark paired error ratios against the two matched comparators, and Table 9 against the exact nonlinear error network (Appendix E) and the finite-difference grids (Appendix F). Against the matched comparators, ratios above one occur only for continued training on Burgers (4.73) and convection (1.13); of these two only the Burgers deficit is resolved at seven runs, by the intervals of Table 10.

Interval estimates for the paired comparisons. Tables 8 and 28 report point ratios and the spread of the paired $\log_{10}$ ratio. Table 10 adds the two-sided 95% t interval on the same paired $\log_{10}$ ratios, back-transformed. These are individual-comparison intervals without family-wise correction across the nine benchmarks and three comparators. Three of them contain one and therefore do not separate the arms at seven runs. Against the matched neural error field, convection gives 0.646 with interval $[0.39, 1.07]$, leaving its direction unresolved at seven runs; the other eight benchmarks exclude one. Against the zero-mean direct solver, heat gives 1.42 with interval $[0.824, 2.43]$, leaving heat unresolved. The intervals favor the direct solver on wave, Poisson 2D, Helmholtz 2D and convection, and favor the correction on Poisson 1D and reaction–diffusion. Against matched-time continued training, convection gives 1.13 with interval $[0.853, 1.51]$ and is likewise undetermined, while the Burgers deficit, 4.73 with interval $[3.08, 7.29]$, is resolved.

Sensitivity to the calibration seed. The training budgets were calibrated on seed 0 against the reference solution (Appendix C), so seed 0 is the one run whose budget was chosen with knowledge of the truth. Table 11 recomputes the geometric-mean improvement with that run removed. The largest change is on Allen–Cahn, from 8.664 to 6.685; convection rises from 2.359 to 2.588, and the remaining benchmarks move by at most 12%. No benchmark changes sign or crosses one. All runs retain the same reference-calibrated budget; the six-run calculation measures sensitivity to the calibration seed within that design.

Training recipe. Table 12 applies the correction to networks trained by vanilla collocation, residual-adaptive sampling and causal weighting. The improvement exceeds one under every applicable recipe; residual-adaptive training lowers it on Helmholtz 2D and weakens the field correlation there. Causal weighting is undefined for the two steady benchmarks.

Frozen-network quality. Table 13 varies the training budget of the frozen network from 500 to $80{,}000$ steps on five benchmarks. The improvement exceeds one at every budget and is largest for networks trained for at most $2{,}000$ steps on Helmholtz 2D, convection and Burgers.

Table 7: Canonical benchmarks, mean ± sample standard deviation over seven independent runs. (a) Final relative L^2 error of the operator-kernel correction, the matched neural error-field control and matched-time continued training. (b) Raw PINN error and, for the operator-kernel correction, the geometric-mean improvement $10^{\bar{d}}$ from the mean paired $\log_{10}$ reduction, the correlation between inferred and true error fields, and the correction wall-clock in seconds. *Corrected error below 10^{-8} in every run (Poisson 1D) or in one run (reaction–diffusion 1D).

(a) Final relative L^2 error of each post-processing arm

Benchmark	Operator-kernel	Neural error field	Continued training
Poisson 1D*	$(4.84 \pm 5.17) \times 10^{-10}$	$(1.96 \pm 1.58) \times 10^{-3}$	$(1.89 \pm 1.02) \times 10^{-4}$
Reaction–diffusion 1D*	$(1.96 \pm 0.58) \times 10^{-8}$	$(5.14 \pm 2.61) \times 10^{-3}$	$(2.40 \pm 0.86) \times 10^{-3}$
Heat 1D	$(1.22 \pm 0.86) \times 10^{-7}$	$(3.49 \pm 0.44) \times 10^{-3}$	$(2.18 \pm 0.20) \times 10^{-3}$
Wave 1D	$(3.92 \pm 2.57) \times 10^{-6}$	$(8.60 \pm 2.02) \times 10^{-3}$	$(6.55 \pm 1.26) \times 10^{-3}$
Poisson 2D	$(8.54 \pm 2.79) \times 10^{-4}$	$(7.43 \pm 1.23) \times 10^{-3}$	$(7.01 \pm 1.24) \times 10^{-3}$
Helmholtz 2D	$(4.14 \pm 1.54) \times 10^{-3}$	$(1.42 \pm 0.58) \times 10^{-2}$	$(1.01 \pm 0.55) \times 10^{-2}$
Convection 1D	$(3.84 \pm 2.46) \times 10^{-3}$	$(5.49 \pm 2.03) \times 10^{-3}$	$(3.50 \pm 2.58) \times 10^{-3}$
Allen–Cahn 1D	$(2.04 \pm 1.41) \times 10^{-3}$	$(1.15 \pm 0.34) \times 10^{-2}$	$(7.90 \pm 2.09) \times 10^{-3}$
Burgers 1D	$(2.24 \pm 1.20) \times 10^{-3}$	$(6.74 \pm 2.91) \times 10^{-3}$	$(4.51 \pm 1.36) \times 10^{-4}$

(b) Raw PINN error and operator-kernel correction

Benchmark	Raw PINN rel. L^2	Improvement	Error-field correlation	Time (s)
Poisson 1D*	$(4.42 \pm 4.64) \times 10^{-3}$	8.349×10^6	$1.000 \pm <0.001$	7.7 ± 1.3
Reaction–diffusion 1D*	$(4.98 \pm 2.62) \times 10^{-3}$	2.345×10^5	$1.000 \pm <0.001$	13.5 ± 0.2
Heat 1D	$(4.42 \pm 1.94) \times 10^{-3}$	4.217×10^4	$1.000 \pm <0.001$	13.7 ± 0.4
Wave 1D	$(1.40 \pm 1.33) \times 10^{-2}$	3511	$1.000 \pm <0.001$	16.4 ± 0.9
Poisson 2D	$(7.78 \pm 1.48) \times 10^{-3}$	9.439	0.995 ± 0.003	13.8 ± 0.7
Helmholtz 2D	$(1.86 \pm 0.92) \times 10^{-2}$	4.252	0.969 ± 0.025	14.8 ± 2.2
Convection 1D	$(8.72 \pm 5.04) \times 10^{-3}$	2.359	0.696 ± 0.359	8.7 ± 0.5
Allen–Cahn 1D	$(1.29 \pm 0.43) \times 10^{-2}$	8.664	0.992 ± 0.009	61.3 ± 1.9
Burgers 1D	$(6.70 \pm 2.95) \times 10^{-3}$	2.971	0.850 ± 0.249	69.9 ± 3.5

Table 8: Per-benchmark paired comparison with the two matched comparators: geometric-mean error ratio (operator-kernel / comparator), paired within each of the seven independent runs, and the standard deviation of the paired $\log_{10}$ ratio. Ratios below one favour the operator-kernel correction; †marks ratios above one, where the comparator is more accurate. Ratios are not averaged across benchmarks.

	Ours / matched neural field		Ours / continued training	
Benchmark	ratio	s.d. $\log_{10}$	ratio	s.d. $\log_{10}$
Poisson 1D	2.15×10^{-7}	0.63	1.70×10^{-6}	0.56
Reaction–diffusion 1D	4.05×10^{-6}	0.20	8.23×10^{-6}	0.22
Heat 1D	2.84×10^{-5}	0.29	4.52×10^{-5}	0.30
Wave 1D	3.73×10^{-4}	0.34	4.88×10^{-4}	0.38
Poisson 2D	0.111	0.20	0.117	0.19
Helmholtz 2D	0.294	0.23	0.423	0.27
Convection 1D	0.646	0.24	$1.13^\dagger$	0.13
Allen–Cahn 1D	0.128	0.36	0.183	0.53
Burgers 1D	0.333	0.31	$4.73^\dagger$	0.20

Figure 2 shows the absolute errors behind those ratios: the uncorrected and corrected relative L^2 error at each base budget. The two behave differently by benchmark. Where the corrected error is well above the observed numerical floor it falls with the base budget — by $69.9\times$ on Burgers, $9.0\times$ on Helmholtz 2D and $4.2\times$ on convection between the smallest and largest budget, though only Helmholtz is monotone across all eight rungs — while the uncorrected error falls faster still, by

Table 9: Per-benchmark paired geometric-mean error ratios (operator-kernel / comparator) over seven independent runs, against the exact nonlinear error network, which applies to the nonlinear benchmarks, and against finite-difference grids with n points per axis, which apply to the linear benchmarks; dashes mark comparisons that do not apply. †Ratio above one, where the comparator is more accurate. The wave benchmark is omitted because its finite-difference comparator diverges under refinement (Appendix F).

	Exact nonlinear	Finite-difference grid			
Benchmark	network	n=17	n=33	n=65	n=129
Poisson 1D	–	1.42×10^{-6}	8.91×10^{-6}	2.98×10^{-5}	1.15×10^{-4}
Reaction–diffusion 1D	–	8.77×10^{-5}	3.42×10^{-4}	1.36×10^{-3}	5.42×10^{-3}
Heat 1D	–	2.78×10^{-4}	1.09×10^{-3}	4.33×10^{-3}	0.0173
Poisson 2D	–	0.824	3.30†	13.0†	52.3†
Helmholtz 2D	–	0.516	5.32†	22.0†	86.0†
Convection 1D	–	0.599	1.01†	1.36†	1.42†
Allen–Cahn 1D	0.127	–	–	–	–
Burgers 1D	0.345	–	–	–	–

Table 10: Two-sided 95% t intervals on the paired $\log_{10}$ error ratios of Tables 8 and 28, back-transformed, over seven independent runs. *Interval contains one. Dashes mark the nonlinear benchmarks, where the linear direct solver does not apply. No family-wise correction is applied.

Benchmark	Ours / neural field	Ours / continued training	Ours / direct solver
Poisson 1D	$[5.6\times10^{-8},\ 8.3\times10^{-7}]$	$[5.1\times10^{-7},\ 5.6\times10^{-6}]$	[0.0245, 0.21]
Reaction–diffusion 1D	$[2.6\times10^{-6},\ 6.2\times10^{-6}]$	$[5.2\times10^{-6},\ 1.3\times10^{-5}]$	[0.2, 0.413]
Heat 1D	$[1.5\times10^{-5},\ 5.3\times10^{-5}]$	$[2.4\times10^{-5},\ 8.6\times10^{-5}]$	[0.824, 2.43]*
Wave 1D	$[1.8\times10^{-4},\ 7.8\times10^{-4}]$	$[2.2\times10^{-4},\ 1.1\times10^{-3}]$	[1.83, 8.73]
Poisson 2D	[0.0717, 0.17]	[0.0779, 0.177]	[462, 954]
Helmholtz 2D	[0.18, 0.48]	[0.238, 0.754]	[845, 2100]
Convection 1D	[0.39, 1.07]*	[0.853, 1.51]*	[34.3, 492]
Allen–Cahn 1D	[0.0599, 0.276]	[0.0589, 0.568]	—
Burgers 1D	[0.174, 0.637]	[3.08, 7.29]	—

Table 11: Geometric-mean improvement of the operator-kernel correction over all seven runs and with the calibration seed removed, and the ratio of the two.

Benchmark	Improvement, 7 runs	Improvement, seed 0 dropped	Ratio
Poisson 1D	8.349×10^{6}	8.217×10^{6}	0.98
Reaction–diffusion 1D	2.345×10^{5}	2.166×10^{5}	0.92
Heat 1D	4.217×10^{4}	3.716×10^{4}	0.88
Wave 1D	3511	3257	0.93
Poisson 2D	9.439	9.298	0.99
Helmholtz 2D	4.252	4.153	0.98
Convection 1D	2.359	2.588	1.10
Allen–Cahn 1D	8.664	6.685	0.77
Burgers 1D	2.971	2.818	0.95

$155\times$, $34.6\times$ and $151\times$. Where the corrected error has reached the floor it is insensitive to the base budget: Poisson 1D ranges from 2.18×10^{-10} to 1.15×10^{-9} and heat 1D stays near 10^{-7} throughout, while their uncorrected errors improve by $3.4\times$ and $5.8\times$. The improvement ratio combines changes in both the raw and corrected error; the corrected mean itself can also increase between successive budgets, as it does from 500 to 1,000 steps on convection and from 2,000 to 5,000 on Burgers. This sweep measures correction quality as the base network receives more training. The records cover the correction arm and contain accuracy measurements; Table 29 separately compares alternatives by additional wall-clock budget.

Table 12: Correction under three PINN training recipes: raw and corrected relative L^2 error, geometric-mean improvement and error-field correlation, mean $\pm$ sample standard deviation over seven independent runs. Causal weighting is not applicable to steady problems.

Recipe	Raw rel. L^2	Corrected rel. L^2	Improvement	Error-field correlation
Poisson 1D				
vanilla	$(4.42 \pm 4.64) \times 10^{-3}$	$(5.09 \pm 3.97) \times 10^{-10}$	6.462×10^{6}	$1.000 \pm <0.001$
residual-adaptive	$(1.95 \pm 3.18) \times 10^{-3}$	$(2.06 \pm 1.90) \times 10^{-10}$	3.700×10^{6}	$1.000 \pm <0.001$
causal weighting	*not applicable to a steady problem*			
Helmholtz 2D				
vanilla	$(1.86 \pm 0.92) \times 10^{-2}$	$(4.14 \pm 1.54) \times 10^{-3}$	4.251	0.969 ± 0.025
residual-adaptive	$(1.14 \pm 0.91) \times 10^{-2}$	$(4.52 \pm 2.14) \times 10^{-3}$	2.300	0.629 ± 0.422
causal weighting	*not applicable to a steady problem*			
Burgers 1D				
vanilla	$(6.70 \pm 2.95) \times 10^{-3}$	$(2.24 \pm 1.20) \times 10^{-3}$	2.971	0.850 ± 0.249
residual-adaptive	$(6.40 \pm 2.81) \times 10^{-3}$	$(2.23 \pm 1.08) \times 10^{-3}$	2.814	0.832 ± 0.260
causal weighting	$(6.38 \pm 2.75) \times 10^{-3}$	$(2.85 \pm 1.11) \times 10^{-3}$	2.199	0.771 ± 0.242

Table 13: Geometric-mean improvement $10^{\bar{d}}$ of the operator-kernel correction for frozen networks trained with different step budgets, over seven independent runs per cell.

Training steps	Poisson 1D	Heat 1D	Helmholtz 2D	Convection 1D	Burgers 1D
500	7.537×10^{6}	2.772×10^{5}	14.66	90.93	9.235
1,000	5.657×10^{6}	1.363×10^{5}	6.868	35.58	10.15
2,000	3.454×10^{6}	1.072×10^{5}	4.629	21.95	11.14
5,000	1.927×10^{6}	6.598×10^{4}	3.409	1.754	2.285
10,000	7.036×10^{6}	9.481×10^{4}	2.787	1.198	2.681
20,000	5.299×10^{6}	5.884×10^{4}	2.653	1.182	2.175
40,000	5.599×10^{6}	5.508×10^{4}	3.042	1.480	2.496
80,000	6.184×10^{6}	3.152×10^{4}	3.237	1.821	3.576

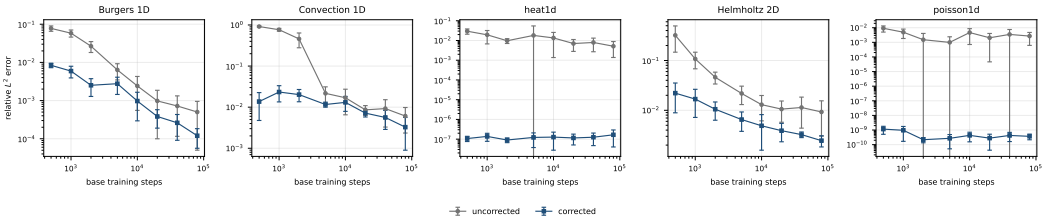


Figure 2: Uncorrected and corrected relative L^2 error against the base training budget of the frozen network, mean $\pm$ sample standard deviation over seven runs per cell, on the five benchmarks the recorded sweep covers. Both axes are logarithmic.

Implementation ablations. Table 14 varies one implementation factor at a time on four benchmarks, refitting the remaining kernel hyperparameters in each variant. Fixing the amplitude replaces the profiled objective by the fixed-amplitude marginal likelihood while the lengthscales and σ_{rel} are still optimized; fixing the lengthscales leaves only σ_{rel} to be optimized, with the amplitude profiled; and only the uniform-placement variant changes the observation sites. Fixing the lengthscales or using float32 arithmetic lowers the improvement on all four. A single global diagonal in place of block-scaled regularization lowers it on Poisson 2D, Helmholtz 2D and convection, but not on Burgers. Omitting a required boundary or initial-condition block can reduce the improvement below one, so that the corrected error exceeds the raw error.

Table 14: Implementation ablations, one factor at a time: final relative L^2 error (mean $\pm$ sample standard deviation over seven independent runs) and geometric-mean improvement against the raw PINN. An improvement below one means the corrected error exceeds the raw error. *Default* rows use the configuration of the main experiments.

Benchmark	Variant	Final rel. L^2	Improvement
Poisson 2D	*default*	$(8.54 \pm 2.79) \times 10^{-4}$	9.437
	fixed lengthscales	$(5.27 \pm 2.90) \times 10^{-3}$	1.838
	fixed amplitude	$(2.65 \pm 2.95) \times 10^{-3}$	4.786
	single global diagonal	$(6.54 \pm 2.64) \times 10^{-3}$	1.354
	uniform instead of Sobol sites	$(3.48 \pm 1.10) \times 10^{-3}$	2.284
	float32 arithmetic	$(2.22 \pm 1.06) \times 10^{-3}$	3.755
	omit block `x_lower_value`	$(6.90 \pm 1.51) \times 10^{-3}$	1.135
	omit block `x_upper_value`	$(4.88 \pm 11.11) \times 10^{-2}$	0.6590
	omit block `y_lower_value`	$(2.19 \pm 3.00) \times 10^{-2}$	0.6393
	omit block `y_upper_value`	$(1.78 \pm 2.94) \times 10^{-2}$	0.8084
Helmholtz 2D	*default*	$(4.14 \pm 1.54) \times 10^{-3}$	4.251
	fixed lengthscales	$(1.49 \pm 0.71) \times 10^{-2}$	1.223
	fixed amplitude	$(1.05 \pm 0.48) \times 10^{-2}$	1.694
	single global diagonal	$(5.94 \pm 1.16) \times 10^{-3}$	2.842
	uniform instead of Sobol sites	$(1.14 \pm 0.25) \times 10^{-2}$	1.479
	float32 arithmetic	$(8.08 \pm 4.49) \times 10^{-3}$	2.307
	omit block `x_lower_value`	$(1.23 \pm 0.63) \times 10^{-2}$	1.519
	omit block `x_upper_value`	$(1.35 \pm 0.44) \times 10^{-2}$	1.277
	omit block `y_lower_value`	$(2.58 \pm 2.16) \times 10^{-2}$	0.9417
	omit block `y_upper_value`	$(2.10 \pm 1.30) \times 10^{-2}$	0.9815
Convection 1D	*default*	$(3.84 \pm 2.46) \times 10^{-3}$	2.359
	fixed lengthscales	$(5.24 \pm 1.98) \times 10^{-3}$	1.600
	fixed amplitude	$(3.66 \pm 2.23) \times 10^{-3}$	2.417
	single global diagonal	$(5.51 \pm 2.12) \times 10^{-3}$	1.551
	uniform instead of Sobol sites	$(3.50 \pm 2.42) \times 10^{-3}$	2.576
	float32 arithmetic	$(4.52 \pm 1.90) \times 10^{-3}$	1.842
	omit block `t_initial_value`	$(4.91 \pm 2.18) \times 10^{-3}$	1.727
	omit block `x_inflow_value`	$(2.71 \pm 4.20) \times 10^{-2}$	0.5478
Burgers 1D	*default*	$(2.24 \pm 1.20) \times 10^{-3}$	2.971
	fixed lengthscales	$(5.59 \pm 2.81) \times 10^{-3}$	1.259
	fixed amplitude	$(6.27 \pm 3.12) \times 10^{-3}$	1.116
	single global diagonal	$(2.08 \pm 1.09) \times 10^{-3}$	3.262
	uniform instead of Sobol sites	$(3.84 \pm 1.28) \times 10^{-3}$	1.669
	float32 arithmetic	$(5.30 \pm 2.36) \times 10^{-3}$	1.273
	omit block `t_initial_value`	$(1.31 \pm 0.46) \times 10^{-2}$	0.4965
	omit block `x_lower_value`	$(2.89 \pm 1.24) \times 10^{-3}$	2.302
	omit block `x_upper_value`	$(1.90 \pm 1.19) \times 10^{-3}$	3.720

E Nonlinear Controls and Newton Diagnostics

The matched exact-nonlinear error-network control tests the complete alternative of optimizing a second neural field against the exact nonlinear error equation, without linearization. On Burgers and Allen–Cahn, a 4×64 tanh network φ is fitted to $N(u_\theta + \varphi) - f = 0$ together with the observed boundary data. The control uses the same frozen network and the same 500 interior and 200 boundary and initial-condition observation sites as the operator-kernel correction, three restarts, the same wall-clock, and checkpoint selection by minimum observable operator-data loss; Table 5 gives its optimization settings.

Table 15 reports the results over seven independent runs. The paired geometric-mean error ratio of the operator-kernel correction to the control is 0.127 on Allen–Cahn and 0.345 on Burgers, so the correction has the lower error on both benchmarks. The error field inferred by the control has mean

correlations of 0.282 ± 0.360 and 0.170 ± 0.101 with the true error. The two procedures differ in function class, regularization, optimization and use of linearization. The measured ratios compare these two complete reconstruction procedures under the stated observation and time budgets.

Table 15: Exact nonlinear error-network control on the two nonlinear benchmarks over seven independent runs: final relative L^2 error of the operator-kernel correction and of the control (mean $\pm$ sample standard deviation), paired geometric-mean error ratio (operator-kernel / control) with the standard deviation of its $\log_{10}$, and the control's error-field correlation. Ratios below one favour the operator-kernel correction.

Benchmark	Operator-kernel final rel. L^2	Exact nonlinear final rel. L^2	Ours / control ratio	s.d. $\log_{10}$ ratio	Control correlation
Allen–Cahn 1D	$(2.04 \pm 1.41) \times 10^{-3}$	$(1.16 \pm 0.34) \times 10^{-2}$	0.127	0.36	0.282 ± 0.360
Burgers 1D	$(2.24 \pm 1.20) \times 10^{-3}$	$(6.55 \pm 2.90) \times 10^{-3}$	0.345	0.31	0.170 ± 0.101

E.1 Resolved Newton Reference and the Error Decomposition

The nonlinear correction of Section 3 makes two approximations at once: it replaces the true error by the solution of the linearized defect problem, and it reconstructs that solution from finitely many scattered operator observations. To separate them we solve the linearized problem directly.

The reference solves the complete linearized boundary-value problem: $N'(u_\theta)\delta = -r$ in the interior together with $\delta = -r_{\partial,j}$ on every Dirichlet and initial face, where $r_{\partial,j}$ is the network's observed defect on that face. It is discretized with second-order central differences in space and Crank–Nicolson in time, one tridiagonal solve per time level, in float64. The linearized coefficients are read from the same automatic extraction the correction itself uses, so the reference and the estimator differentiate the same operator; on Burgers the extracted coefficients agree with the closed-form linearization to machine zero.

Grids are chosen as $n = 199m + 1$ per axis so that every one of the 200 fixed evaluation nodes per axis coincides with a reference node. The reference field is then read by indexing and is never interpolated onto the evaluation grid. The refinement ladder is $m \in \{4, 8, 16\}$, with $m = 32$ held in reserve. The accuracy criteria were fixed before running: successive-refinement relative difference at most 10^{-3} with observed order in $[1.5, 2.5]$, and an independent Crank–Nicolson versus BDF2 discrepancy at most 5×10^{-3}.

Table 16: Reference Newton convergence at the accepted resolution $m = 16$. All four cases met the predeclared criteria; $m = 32$ was not required. The observed order is measured on the $m \in \{4, 8, 16\}$ pilot of each case.

Case	n per axis	Refinement diff.	Order p	CN vs. BDF2	Criteria
Allen–Cahn canonical	3185	0.34×10^{-4}	2.000	8.77×10^{-5}	passed
Burgers canonical	3185	3.64×10^{-4}	2.001	1.14×10^{-5}	passed
Burgers held-out 016	3185	3.50×10^{-4}	2.007	0.05×10^{-5}	passed
Burgers held-out 017	3185	2.32×10^{-4}	2.002	0.60×10^{-5}	passed

The *ideal Newton correction* δ solves the linearized defect problem exactly; the *resolved Newton correction* δ_{ref} is its grid-resolved numerical approximation. The estimator reconstructs $\widehat{e}$ from 700 observations. Proposition 2 conditionally bounds the linearization gap $e - \delta$ at second order. The remaining reconstruction gap $\delta - \widehat{e}$ is measured below. With $e = u - u_\theta$ and $\widehat{e}$ the operator-kernel correction, define

$$L = \frac{\|e - \delta_{\mathrm{ref}}\|}{\|e\|}, \qquad R = \frac{\|\delta_{\mathrm{ref}} - \widehat{e}\|}{\|e\|}, \qquad T = \frac{\|e - \widehat{e}\|}{\|e\|}, \tag{38}$$

all in the fixed evaluation measure of the benchmark. Since $e - \widehat{e} = (e - \delta_{\mathrm{ref}}) + (\delta_{\mathrm{ref}} - \widehat{e})$, the identity that holds is $T^2 = L^2 + R^2 + 2LR\cos\angle$ with $\cos\angle$ the cosine between the two components; T is not in general $L + R$. The cosine is reported as a cancellation diagnostic. The labels *reconstruction-dominated* ($R \geq 3L$), *linearization-dominated* ($L \geq 3R$) and *mixed* were fixed before the runs and are descriptive only.

Table 17: Newton decomposition of equation 38, mean ± sample standard deviation. Canonical rows are over seven runs; each held-out row is over the three repeated runs of that named physical instance, and the two instances are not pooled.

Case	n	L	R	T	$\cos\angle$	Label
Allen–Cahn	7	0.0559 ± 0.0647	0.1488 ± 0.0653	0.1440 ± 0.0824	-0.22	mixed across runs
Burgers	7	0.0137 ± 0.0091	0.4147 ± 0.2967	0.4168 ± 0.2934	$+0.01$	reconstruction-dom.
Burgers h/o 016	3	0.0053 ± 0.0027	0.9830 ± 0.0219	0.9806 ± 0.0211	-0.39	reconstruction-dom.
Burgers h/o 017	3	0.0071 ± 0.0024	0.7677 ± 0.1806	0.7666 ± 0.1806	-0.14	reconstruction-dom.

Table 17 gives the outcome. On the three Burgers cases R exceeds $3L$ on every run: the resolved Newton step is close to the true error in relative norm and the reconstruction carries the remaining discrepancy, most sharply on held-out instance 016, whose layer is the thinnest in the family and where R approaches unity. Allen–Cahn is mixed: its per-run labels are split between reconstruction-dominated and mixed, and one run has $L = 0.17$ and $R = 0.15$ combine to $T = 0.07$ at $\cos\angle = -0.93$, a near cancellation that would be misread by treating T as $L + R$. The identity above closes to at most 2.1×10^{-17} on every run.

E.2 Batlle-Style Nonlinear Kernel Solver

Section 2 notes that nonlinear kernel solvers proceed by successive linearizations. To place the post-hoc correction against that route we implemented a comparator based on Algorithm 2.2 of Batlle et al. (2025) and ran it on the two nonlinear benchmarks.

Our implementation of the published algorithm uses the solution itself as the unknown in the reproducing kernel Hilbert space; the interior and boundary functionals $\delta(s_m) \circ P'(u_{\ell-1})$ and $\delta(s_m) \circ B'(u_{\ell-1})$ enter one Gram system of order equal to the number of collocation points; the iterate starts at $u_0 = 0$; steps are full, with no line search; the Gram is rebuilt and refactorized at every nonlinear iteration; the lengthscale is a single isotropic value and is never fitted; and the regularization is the adaptive nugget $\eta \operatorname{diag}(K)$ of the source rather than ηI. Exactly three nonlinear iterations are used, matching the setting reported in the source. The observation budget is matched to the correction at 500 interior and 200 boundary functionals, giving a Gram of order 700.

The lengthscale and nugget are prescribed rather than fitted, so a configuration was selected on seed 0 alone from a grid of five lengthscales and five nuggets fixed before any run. Selection minimized the root mean square of the PDE residual and the boundary defect on 125 interior and 50 boundary validation points independent of the collocation sites; the reference solution was not evaluated during selection. The selected configurations, $(\ell, \eta) = (0.40, 10^{-3})$ for Allen–Cahn and $(0.40, 10^{-10})$ for Burgers, were then frozen for all seven runs. All 50 swept configurations and all 14 evaluation runs completed with finite output and successful factorization.

At each iteration, we evaluate $P'(u_{\ell-1})$ at the collocation sites and use these coefficient values in the Gram and cross-covariance assembly. The right-hand side uses the same linearization. This assembles the prescribed discrete system without repeatedly differentiating through the previous 700-term kernel expansion; a solve takes about thirty seconds.

Table 18: Nonlinear kernel solver based on Batlle et al. (2025) Algorithm 2.2, at a matched budget of 700 functionals and three nonlinear iterations, against the corrected PINN. Mean ± sample standard deviation over seven runs. Runtimes are per solve and exclude the one-off 958 s configuration sweep. The comparison to the PINN arms is unpaired: the solver shares no frozen network or site realization with them.

Benchmark	(ℓ, η)	Batlle rel. L^2	Corrected rel. L^2	Batlle solve	PINN + corr.
Allen–Cahn	$(0.40,\ 10^{-3})$	$(8.27 \pm 0.60) \times 10^{-1}$	$(2.04 \pm 1.41) \times 10^{-3}$	31 s	107 s
Burgers	$(0.40,\ 10^{-10})$	$(1.44 \pm 0.19) \times 10^{-2}$	$(2.24 \pm 1.20) \times 10^{-3}$	30 s	93 s

At this budget the solver reaches a relative L^2 error of 8.27×10^{-1} on Allen–Cahn and 1.44×10^{-2} on Burgers, against 2.04×10^{-3} and 2.24×10^{-3} for the corrected PINN, and 7.90×10^{-3} and

4.52×10^{-4} for matched-time continued training. Its solve takes about thirty seconds, compared with recorded full-pipeline costs of 107 s and 93 s for training and correcting a PINN on the two benchmarks. These timings come from separate executions at the stated configurations. The accuracy comparison fixes the observation count and three nonlinear iterations; the experiments of Batlle et al. (2025) use one to eight thousand interior collocation points. Together, the reported errors and solve times describe the tradeoff at this fixed discretization and iteration budget.

E.3 Prior Mean and Regularization Ablation

The nonlinear construction combines the frozen-network solution-space prior mean u_θ with the Stage-2 regularization rule of equation 37. We isolate their effects in a 2×2 design on the two canonical nonlinear benchmarks, using the same seven frozen networks per benchmark as Table 17. Cell A is a single fitted stage at prior mean u_θ; cell B adds the second stage and is exactly the method reported throughout the paper; cell C is a single stage at prior mean 0; cell D is the two-stage rule at prior mean 0.

All cells retain the same frozen networks, collocation and evaluation sites, kernel family, length-scales, relative nugget, jitter, block scaling, and constant C_N from the original runs. Within either stage the two mean arms use the same Λ, kernel and actual Σ, including the same profiled amplitude. In particular, Stage 2 is not re-estimated for the zero-mean arm, which inherits the rule of the u_θ-mean pipeline. The two predictions are $\hat{u}_{u_\theta} = u_\theta + \mathcal{S}y$ and $\hat{u}_0 = \mathcal{S}(\Lambda u_\theta + y)$; hence $\hat{u}_{u_\theta} - \hat{u}_0 = u_\theta - \mathcal{S}(\Lambda u_\theta)$, where $\mathcal{S}$ is the conditioning map of that cell; it was checked at every run and held to 2.3×10^{-11}. Re-running cell B reproduced the archived corrected errors to a maximum relative difference of 9.4×10^{-13} over all fourteen runs, and reproduced the decomposition of Table 17 to the digits printed there, so the other three cells differ from the published pipeline only in the factor being varied.

(a) Final relative L^2 error per cell

PDE	Cell	Solution prior mean	Stage	Final rel. L^2
Allen–Cahn	A	u_θ	1	$(8.98 \pm 11.68) \times 10^{-4}$
	B (paper's method)	u_θ	2	$(2.04 \pm 1.41) \times 10^{-3}$
	C	0	1	$(1.01 \pm 1.33) \times 10^{-3}$
	D	0	2	$(7.64 \pm 7.60) \times 10^{-2}$
Burgers	A	u_θ	1	$(2.24 \pm 1.19) \times 10^{-3}$
	B (paper's method)	u_θ	2	$(2.24 \pm 1.20) \times 10^{-3}$
	C	0	1	$(2.68 \pm 2.74) \times 10^{-1}$
	D	0	2	$(2.70 \pm 2.72) \times 10^{-1}$

(b) Paired ratios between cells

PDE	Paired ratio (direction as written)	Geometric mean	95% interval
Allen–Cahn	$E_{\text{Stage 2}}/E_{\text{Stage 1}}$ at mean u_θ	3.28	[1.06, 10.1]
	$E_{\text{Stage 2}}/E_{\text{Stage 1}}$ at mean 0	108	[33, 354]
	$E_{\text{mean 0}}/E_{\text{mean } u_\theta}$ at Stage 1	1.05	[0.931, 1.17]
	$E_{\text{mean 0}}/E_{\text{mean } u_\theta}$ at Stage 2	34.5	[18.8, 63.2]
Burgers	$E_{\text{Stage 2}}/E_{\text{Stage 1}}$ at mean u_θ	0.999	[0.978, 1.02]
	$E_{\text{Stage 2}}/E_{\text{Stage 1}}$ at mean 0	1.06	[0.972, 1.15]
	$E_{\text{mean 0}}/E_{\text{mean } u_\theta}$ at Stage 1	70.9	[25.9, 194]
	$E_{\text{mean 0}}/E_{\text{mean } u_\theta}$ at Stage 2	75.1	[29.5, 191]

Table 19: Crossing the solution-space prior mean with the second stage, on the seven original frozen networks per benchmark. Panel (a) gives the final relative L^2 error of each cell as mean $\pm$ sample standard deviation; panel (b) gives the paired ratios as the back-transformed mean of the per-run $\log_{10}$ ratios with a two-sided 95% interval, each descriptive of its own comparison and with no family-wise correction. Ratios above one favour the denominator.

The linearization term L of equation 38 depends only on the frozen network and the PDE, so it is identical in all four cells, at 0.0559 on Allen–Cahn and 0.0137 on Burgers; the design acts entirely on the reconstruction term R.

At prior mean u_θ the second stage changes the Burgers error by a factor of 0.999 with interval $[0.978, 1.02]$, moving the reconstruction term from 0.4145 to 0.4147; on Allen–Cahn it multiplies the final error by 3.28 with interval $[1.06, 10.1]$ and moves the reconstruction term from 0.0053 to 0.1488. On Allen–Cahn the single fitted stage recovers the resolved Newton correction closely, and the enlarged regularization then damps part of what it had recovered.

The two quantities do not move together, and the per-run directions show why. On Allen–Cahn the reconstruction term is larger with the second stage in all seven runs, while the final error is larger in six: on the remaining run the reconstruction term rises from 0.0041 to 0.1462 and the final error nonetheless falls from 3.00×10^{-3} to 1.15×10^{-3}, because the reconstruction discrepancy and the linearization discrepancy partly cancel there, as the cosine term of equation 38 allows. A change in R affects the final error through both its magnitude and its alignment with the linearization component. On Burgers both are larger in six of seven runs, while the paired final-error interval includes one.

At the recorded hyperparameters, the Stage-2 update increases final error on Allen–Cahn; the Burgers interval includes one. These comparisons isolate the regularization update in the complete construction (cell B) against its single-stage ablation (cell A).

The prior mean behaves differently on the two benchmarks, and differently at the two stages. At Stage 1, discarding u_θ costs a factor of 70.9 on Burgers with interval $[25.9, 194]$, with the zero-mean reconstruction term rising from 0.41 to 64.9, so that the zero-mean cell is far worse there than leaving the network uncorrected; on Allen–Cahn the same substitution gives 1.05 with interval $[0.931, 1.17]$, leaving the direction of the Allen–Cahn comparison unresolved across these seven runs.

At Stage 2 the substitution costs 34.5 with interval $[18.8, 63.2]$ on Allen–Cahn and 75.1 with interval $[29.5, 191]$ on Burgers. The Allen–Cahn contrast between 1.05 at Stage 1 and 34.5 at Stage 2 is an interaction between the two factors: the effect of removing the prior mean depends on the regularization stage, and the effect of the second stage depends on the prior mean.

All four cells inherit the hyperparameters fitted with the u_θ prior mean. The comparisons therefore measure changes in prior mean and regularization at common, inherited settings on these two benchmarks. The zero-mean cells retain those settings throughout the ablation.

F Finite-Difference Error Solves

A classical alternative is to discretize the same defect equation and solve it, which is the approach of Krasowski et al. (2026). The comparator discretizes the error equation on a uniform grid with $n = 17$, 33, 65 or 129 points per axis, endpoints included: an n^d grid for the steady problems and an $n \times n$ grid in (x, t) for the time-dependent ones, which are assembled as a single space–time linear system rather than advanced by time stepping. Interior rows use second-order central differences. Boundary values, and initial values at $t = 0$, are imposed by strong Dirichlet rows that replace the row by the identity with the observed error datum on the right-hand side; the wave system adds a row for the initial velocity, and on convection the inflow face carries a strong row while the outflow endpoint uses a second-order backward difference. The sparse system is solved directly in float64 on a CPU, and the grid solution is interpolated to the evaluation points with a regular-grid interpolator. The comparison covers the seven linear benchmarks and uses the operator-kernel correction at its fixed budget of a few hundred scattered operator observations (Table 6). Table 20 gives the errors over seven independent runs and the residual evaluations of the finest grid, and Table 9 gives the paired geometric-mean error ratios of the correction to each grid. This appendix uses the separately recorded finite-difference cohort, including its recomputed kernel correction; Table 7 retains the original canonical cohort.

Smooth one-dimensional problems. On Poisson 1D, and on heat and reaction–diffusion with their $n \times n$ space–time grids, the correction remains more accurate than the grid throughout the sweep: at $n = 129$ the paired ratios of the correction to the grid are 1.15×10^{-4}, 0.0173 and 5.42×10^{-3}. On these problems the mean corrected error lies between 10^{-10} and 10^{-7} (Table 7), whereas the error of the second-order grid decreases only algebraically under refinement.

Regular two-dimensional problems and convection. On Poisson 2D and Helmholtz 2D the correction is more accurate than the $n = 17$ grid, but the grid is more accurate from $n = 33$, with ratios of 3.30 and 5.32, rising to 52.3 and 86.0 at $n = 129$, where it uses roughly 17,000 residual evaluations. The grid uses tensor-product structure and boundary data at every boundary node, whereas the kernel estimator uses a few hundred scattered observations. On convection, the paired point ratio is 1.01 at $n = 33$ and 1.36 at $n = 65$, both favouring the grid. These point ratios are not equivalence tests or interval estimates.

Wave comparator. For the wave benchmark, the error of the tested central space–time discretization grows rapidly under refinement, to a mean relative error of order 10^{68} at $n = 129$. This is a numerical failure of the tested discretization, not an inapplicable PDE. All seven runs at each resolution remain in Table 20; the unstable comparator is excluded from accuracy-ranking summaries.

Table 20: Finite-difference solves of the same error equation with n points per axis: final relative L^2 error, mean $\pm$ sample standard deviation over seven independent runs, with the raw PINN and the operator-kernel correction in panel (a). The last column gives the residual evaluations of the $n = 129$ grid. †The wave comparator diverges under refinement; all its runs are retained in this table and the failed comparator is excluded from accuracy-ranking summaries.

(a) Raw PINN, correction and the coarsest grid

Benchmark	Raw PINN	Operator-kernel	Grid n=17
Poisson 1D	$(4.42 \pm 4.64) \times 10^{-3}$	$(4.85 \pm 5.18) \times 10^{-10}$	$(2.60 \pm 2.33) \times 10^{-4}$
Heat 1D	$(4.42 \pm 1.94) \times 10^{-3}$	$(1.22 \pm 0.86) \times 10^{-7}$	$(3.57 \pm 0.55) \times 10^{-4}$
Reaction–diffusion 1D	$(4.98 \pm 2.62) \times 10^{-3}$	$(1.96 \pm 0.58) \times 10^{-8}$	$(2.24 \pm 0.89) \times 10^{-4}$
Poisson 2D	$(7.78 \pm 1.48) \times 10^{-3}$	$(8.54 \pm 2.79) \times 10^{-4}$	$(9.95 \pm 1.53) \times 10^{-4}$
Helmholtz 2D	$(1.86 \pm 0.92) \times 10^{-2}$	$(4.14 \pm 1.54) \times 10^{-3}$	$(7.84 \pm 2.42) \times 10^{-3}$
Convection 1D	$(8.72 \pm 5.04) \times 10^{-3}$	$(3.84 \pm 2.46) \times 10^{-3}$	$(5.78 \pm 1.69) \times 10^{-3}$
Wave 1D†	$(1.40 \pm 1.33) \times 10^{-2}$	$(3.92 \pm 2.57) \times 10^{-6}$	$(5.44 \pm 1.50) \times 10^{10}$

(b) Refined grids

Benchmark	Grid n=33	Grid n=65	Grid n=129	Evaluations at n=129
Poisson 1D	$(4.94 \pm 4.68) \times 10^{-5}$	$(1.27 \pm 1.05) \times 10^{-5}$	$(3.20 \pm 2.55) \times 10^{-6}$	131
Heat 1D	$(9.12 \pm 1.38) \times 10^{-5}$	$(2.29 \pm 0.34) \times 10^{-5}$	$(5.74 \pm 0.86) \times 10^{-6}$	17,026
Reaction–diffusion 1D	$(5.72 \pm 2.18) \times 10^{-5}$	$(1.44 \pm 0.54) \times 10^{-5}$	$(3.60 \pm 1.35) \times 10^{-6}$	17,026
Poisson 2D	$(2.49 \pm 0.42) \times 10^{-4}$	$(6.30 \pm 1.02) \times 10^{-5}$	$(1.57 \pm 0.25) \times 10^{-5}$	17,153
Helmholtz 2D	$(7.78 \pm 3.24) \times 10^{-4}$	$(1.90 \pm 0.81) \times 10^{-4}$	$(4.88 \pm 2.08) \times 10^{-5}$	17,153
Convection 1D	$(3.57 \pm 1.73) \times 10^{-3}$	$(2.73 \pm 1.65) \times 10^{-3}$	$(2.46 \pm 0.92) \times 10^{-3}$	16,898
Wave 1D†	$(1.92 \pm 1.11) \times 10^{22}$	$(3.59 \pm 2.24) \times 10^{40}$	$(2.29 \pm 2.00) \times 10^{68}$	17,153

Scope of the comparison. The comparison uses second-order central differences; higher-order or stabilized schemes can be more accurate. The kernel correction is more accurate on the three smooth one-dimensional problems across the tested grids, while refinement gives lower errors on the regular two-dimensional boxes and convection. The residual-evaluation counts refer to this comparison and do not measure computational cost in general.

G Held-Out Instances, Irregular Domain and Three-Dimensional Case

Held-out population construction. Four parameterized families, Poisson, Helmholtz, convection and Burgers, each contain 20 physical instances, 80 in total, generated with the fixed seed 20260905; each instance is run three times. Table 22 lists the parameters varied within each family. The PDE class is fixed within a family, but what varies across its instances is not the same in all four: depending on the family the instance parameters change the operator coefficients, the forcing, or the boundary and initial data, and in every family they change the reference solution. Table 23 gives

Table 21: Held-out physical instances, six per family and 24 overall. The physical instance is the statistical unit: the three runs of each instance are averaged first, then means ± sample standard deviations are taken across instances. Improvement is $10^{\bar{d}}$ from the mean $\log_{10}$ reduction.

Family	Instances	Raw rel. L^2	Corrected rel. L^2	Mean $\log_{10}$ reduction	Improvement
Poisson	6	$(5.88 \pm 5.49) \times 10^{-3}$	$(1.96 \pm 0.45) \times 10^{-9}$	6.00 ± 0.68	1.011×10^{6}
Helmholtz	6	$(3.78 \pm 6.03) \times 10^{-1}$	$(6.31 \pm 14.00) \times 10^{-2}$	1.54 ± 0.76	34.84
Burgers	6	$(8.28 \pm 5.17) \times 10^{-3}$	$(1.58 \pm 2.37) \times 10^{-3}$	1.44 ± 1.18	27.43
Convection	6	$(5.22 \pm 10.46) \times 10^{-2}$	$(4.82 \pm 10.47) \times 10^{-2}$	0.24 ± 0.12	1.728
All held-out	24	$(1.11 \pm 3.26) \times 10^{-1}$	$(2.82 \pm 8.64) \times 10^{-2}$	2.31 ± 2.36	202.1

the mechanism for each. The development/test split was written before any solve was evaluated: instances are assigned within family by the SHA-256 hash of the split seed 20260905 and the instance identifier, the split is balanced by family (14 development and 6 test instances per family, 56 and 24 overall), and all runs of an instance stay on the same side. The only recorded use of the development instances was the selection of acceptance thresholds for a gating rule that is not part of the reported results, and those thresholds were frozen before evaluation. The retained correction pipeline has no cross-instance parameter fitted on the development population; kernel hyperparameters are estimated independently for each frozen solve without reference data. Development and test instances come from the same families, so the test measures generalization across physical instances within known families.

Table 22: Parameters varied across the 20 physical instances of each held-out family. $\mathcal{U}(a, b)$: continuous uniform; $\mathcal{U}\{a, \dots, b\}$: uniform over integers; log-uniform: $\exp \mathcal{U}(\log a, \log b)$.

Family	Varied parameters
Poisson	modes $m_1 \sim \mathcal{U}\{1, \dots, 4\}$ and $m_2 \sim \mathcal{U}\{5, \dots, 9\}$; amplitudes $a_1 \sim \mathcal{U}(0.5, 1.5)$ and $a_2 \sim \mathcal{U}(0.1, 0.5)$; boundary offset $\sim \mathcal{U}(-0.4, 0.4)$; boundary slope $\sim \mathcal{U}(-0.6, 0.6)$
Helmholtz	wavenumber $\sim \mathcal{U}(0.5, 8)$; modes $m_x \sim \mathcal{U}\{1, \dots, 4\}$ and $m_y \sim \mathcal{U}\{1, \dots, 5\}$; amplitude $\sim \mathcal{U}(0.5, 1.5)$; interaction amplitude $\sim \mathcal{U}(0.1, 0.45)$; boundary offset $\sim \mathcal{U}(-0.25, 0.25)$; boundary terms in x and in $y \sim \mathcal{U}(-0.2, 0.2)$
Convection	transport speed $\sim \mathcal{U}(5, 40)$; frequency $\sim \mathcal{U}\{1, 2, 3\}$; amplitude $\sim \mathcal{U}(0.5, 1.5)$; phase $\sim \mathcal{U}(-\pi, \pi)$; offset $\sim \mathcal{U}(-0.3, 0.3)$
Burgers	viscosity log-uniform on $[0.0025, 0.02]$; wave speed $\sim \mathcal{U}(-0.25, 0.65)$; wave amplitude $\sim \mathcal{U}(0.08, 0.22)$; initial layer centre $\sim \mathcal{U}(-0.45, 0.25)$

All four families share one generation mechanism: the sampled physical parameters define a closed-form reference field u_p; the specified differential operator is applied to u_p by automatic differentiation to produce the forcing; and the boundary and initial data are traces of u_p on the faces the family declares. Two families depart from it, and the departure is the same one: for convection and Burgers the forcing is declared identically zero rather than manufactured, because their reference fields solve the homogeneous equation exactly. Table 23 states each family in full.

Aggregation. The physical instance is the statistical unit. The three runs of each instance are averaged first, giving one raw error, one corrected error and one $\log_{10}$ reduction per instance; means and sample standard deviations are then taken over the six instances of a family, or over all 24 held-out instances for the overall row. The geometric-mean improvement is $10^{\bar{d}}$ for the mean per-instance $\log_{10}$ reduction $\bar{d}$.

Held-out results. Table 21 reports the four families and the overall row, and Table 24 lists every held-out instance. Convection is the weakest family, with a mean $\log_{10}$ reduction of 0.24 ± 0.12. Error-field correlation is not reported for the held-out instances.

Table 23: Held-out family definitions. u_p is the closed-form reference field built from the instance parameters of Table 22; f is the forcing; boundary and initial data are traces of u_p on the listed faces. The Burgers family is the one that carries an initial layer centre x_c; the canonical Burgers benchmark of Appendix C.1 has no such parameter.

Family	Domain, operator, reference field, forcing, conditions
Poisson	$x \in [0,1]$; $-u_{xx} = f$; $u_p = b_0 + b_1 x + a_1 \sin(m_1 \pi x) + a_2 \sin(m_2 \pi x)$; $f = -u_p''$ by automatic differentiation; Dirichlet at both endpoints. *Instances vary the reference field, hence f and both Dirichlet values; the operator is fixed.*
Helmholtz	$(x,y) \in [-1,1]^2$; $\Delta u + k^2 u = f$; $u_p = b_0 + b_x x + b_y y + A \sin\big(m_x \pi \frac{x+1}{2}\big) \sin\big(m_y \pi \frac{y+1}{2}\big) + A_{\text{int}} \sin\big(\pi(x+1)\big) \sin\big(1.5\pi(y+1)\big)$; $f = \Delta u_p + k^2 u_p$ by automatic differentiation; Dirichlet on all four faces. *Instances vary the zeroth-order coefficient: the sampled wavenumber enters the operator as k^2, so $k = 8$ gives a coefficient of 64.*
Convection	$(x,t) \in [0,2\pi] \times [0,1]$; $u_t + \beta u_x = 0$; $u_p = c_0 + A \sin\big(\nu(x - \beta t) + \varphi\big)$; $f \equiv 0$, declared zero, since u_p solves the homogeneous equation; Dirichlet on the inflow face $x = 0$ and initial data at $t = 0$, no outflow condition. *Instances vary the first-order transport coefficient β together with the reference field.*
Burgers	$(x,t) \in [-1,1] \times [0,1]$; $u_t + u u_x - \nu u_{xx} = 0$, nonlinear, rebuilt per instance at that instance's ν; $u_p = C - A \tanh\big(A(x - x_c - Ct)/(2\nu)\big)$, a travelling wave of width $2\nu/A$ centred at $x_c + Ct$; $f \equiv 0$, declared zero; Dirichlet at $x = \pm 1$ and initial data at $t = 0$. *Instances vary the viscosity in the operator itself and the reference field; this family carries the initial layer centre x_c.*

Table 24: Every held-out physical instance; each value is the average over the three runs of that instance.

Family	Instance	Raw rel. L^2	Corrected rel. L^2	$\log_{10}$ reduction
Poisson	poisson_000	4.71×10^{-4}	1.56×10^{-9}	5.25
	poisson_004	7.65×10^{-3}	2.45×10^{-9}	6.59
	poisson_005	8.82×10^{-4}	1.48×10^{-9}	5.53
	poisson_007	3.92×10^{-3}	1.91×10^{-9}	6.21
	poisson_009	7.11×10^{-3}	2.57×10^{-9}	5.50
	poisson_010	1.53×10^{-2}	1.78×10^{-9}	6.94
Helmholtz	helmholtz_000	4.78×10^{-3}	7.67×10^{-4}	0.85
	helmholtz_002	2.04×10^{-3}	1.40×10^{-5}	2.23
	helmholtz_004	7.69×10^{-1}	3.48×10^{-1}	0.60
	helmholtz_009	1.44	2.90×10^{-2}	1.70
	helmholtz_011	1.65×10^{-2}	1.03×10^{-3}	1.33
	helmholtz_015	3.28×10^{-2}	1.35×10^{-4}	2.54
Burgers	burgers_005	1.40×10^{-2}	3.99×10^{-5}	2.57
	burgers_008	4.56×10^{-3}	2.50×10^{-4}	1.27
	burgers_011	5.29×10^{-3}	1.17×10^{-5}	2.67
	burgers_016	5.71×10^{-3}	5.61×10^{-3}	0.01
	burgers_017	4.36×10^{-3}	3.40×10^{-3}	0.12
	burgers_019	1.57×10^{-2}	1.79×10^{-4}	1.99
Convection	convection_002	2.66×10^{-1}	2.62×10^{-1}	0.12
	convection_003	1.29×10^{-2}	9.19×10^{-3}	0.17
	convection_010	1.18×10^{-2}	4.42×10^{-3}	0.41
	convection_013	1.21×10^{-2}	7.39×10^{-3}	0.16
	convection_014	4.01×10^{-3}	1.66×10^{-3}	0.36
	convection_016	6.77×10^{-3}	4.51×10^{-3}	0.21

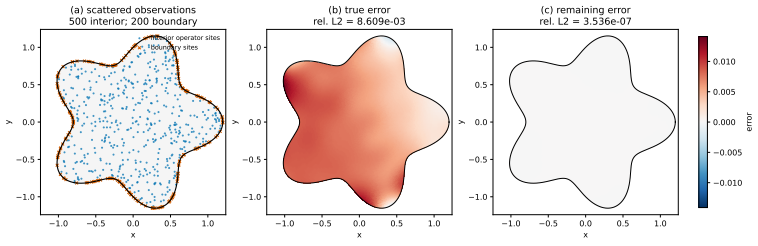


Figure 3: Star-shaped domain. Left: the scattered operator and boundary sites, reconstructed from the recorded run and its deterministic sampling rule (Appendix G). Centre and right: the true PINN error and the remaining error after correction, on a shared colour scale. No Cartesian mesh is constructed.

Irregular domain. The problem is a two-dimensional Poisson problem $-\Delta u = f$ on the star-shaped domain $\Omega = \{(\rho, \phi) : \rho \le r(\phi)\}$ with $r(\phi) = 1 + 0.2\cos(5\phi)$, so the radius runs between 0.8 and 1.2 over five lobes. The manufactured solution is the smooth non-separable field

$$\begin{aligned} u(x,y) = {} & \sin(1.7x + 0.9y + 0.35xy) + 0.28\cos(1.15xy - 0.45x + 0.20y) \\ & + 0.22\exp\big(-6[(x-0.24)^2 + (y+0.18)^2]\big), \end{aligned} \tag{39}$$

defined on a neighbourhood of $\overline{\Omega}$. As in the held-out families the forcing is produced only by applying the specified operator to equation 39 by automatic differentiation, $f = -\Delta u$, with no closed form written down, and the inhomogeneous Dirichlet data is the trace of the same field on the complete boundary. The frozen network is trained for $5{,}000$ steps with the default recipe of Appendix C.2. The correction uses 500 interior sites drawn by uniform-area polar rejection sampling and 200 boundary sites drawn independently in the boundary angle. The sites displayed in Figure 3 are reconstructed from the recorded run seed with the recorded correction-site sampling convention; the original coordinate arrays were not archived. Unlike the canonical suite, the error on this domain is the quadrature-weighted relative L^2 error $\big(\sum_i w_i|u_i - \widehat{u}_i|^2 / \sum_i w_i|u_i|^2\big)^{1/2}$ over the nodes and weights w_i of a polar-area quadrature whose nodes lie strictly inside the domain, because a tensor grid over the bounding box would place nodes where the manufactured field does not solve the problem. Both matched comparators are run under the protocol of Appendix C, and Table 25 gives the four arms. The operator-kernel correction reduces the mean relative error from $(3.25 \pm 3.24) \times 10^{-3}$ to $(8.16 \pm 5.73) \times 10^{-7}$, a geometric-mean improvement of $3353\times$, against $2.279\times$ for the matched neural error-field control and $3.008\times$ for matched-time continued training. The finite-difference comparator of Appendix F requires a Cartesian grid and is therefore not run on this domain; it could be applied after constructing a suitable mesh or embedded-boundary discretization.

Table 25: Irregular star-shaped domain $r(\phi) = 1 + 0.2\cos(5\phi)$, four arms, mean $\pm$ sample standard deviation over seven independent runs. Improvement is $10^{\bar{d}}$ from the mean paired $\log_{10}$ reduction against the raw PINN, which serves as the reference row.

Arm	Final rel. L^2	Mean $\log_{10}$ reduction	Improvement
Raw PINN	$(3.25 \pm 3.24) \times 10^{-3}$	reference	
Operator-kernel correction	$(8.16 \pm 5.73) \times 10^{-7}$	3.52 ± 0.55	3353
Matched neural error-field control	$(1.01 \pm 0.21) \times 10^{-3}$	0.36 ± 0.41	2.279
Matched-time continued training	$(7.72 \pm 2.00) \times 10^{-4}$	0.48 ± 0.39	3.008

Same-site direct kernel solve. The correction-site coordinate arrays of the original star study were not archived, so the sites were reconstructed from the recorded seeds and the deterministic sampling rule. To guarantee that the compared arms see one site realization, the raw network, the operator-kernel correction and a direct zero-mean kernel solve were run together in a single process on the reconstructed sites; the neural error-field and continued-training values quoted above come from the original study and are not part of this rerun. A reconstruction check was performed first: the

reconstructed raw metric reproduces the archived value exactly on all seven seeds, and the corrected mean agrees to 3.7×10^{-5} relative. The common reconstructed sites define the pairing in this comparison.

Both kernel arms use the same squared-exponential family, the same 500 interior and 200 boundary sites and the same polar-area quadrature, and each fits its own hyperparameters. The direct solve is more accurate: the paired geometric-mean error ratio of post-hoc to direct is 5.91 (direct to post-hoc, 0.169), at 18.5 s against 20.3 s of fitting time. Both kernel formulations operate directly on scattered observations in the irregular domain without a Cartesian mesh; the direct solve gives the lower error at comparable fitting cost in this case.

Table 26: Star domain, three arms sharing one reconstructed site realization. Weighted relative L^2 as mean $\pm$ sample standard deviation over seven runs; improvement is $10^{\bar{d}}$ with $\bar{d}$ the mean paired $\log_{10}$ reduction against the raw network on the same run.

Arm (same reconstructed sites)	Weighted rel. L^2	Mean $\log_{10}$ reduction	Improvement
Raw PINN	$(3.25 \pm 3.24) \times 10^{-3}$	reference	
Operator-kernel correction	$(8.16 \pm 5.73) \times 10^{-7}$	3.525 ± 0.552	$3353\times$
Direct zero-mean kernel solve	$(1.18 \pm 0.43) \times 10^{-7}$	4.297 ± 0.432	$19819\times$

Smooth three-dimensional case. The problem is $-\nabla \cdot (D \nabla u) = f$ on $[0,1]^3$ with constant symmetric positive definite

$$D = \begin{pmatrix} 1.00 & 0.12 & 0.08 \\ 0.12 & 0.80 & 0.06 \\ 0.08 & 0.06 & 0.65 \end{pmatrix}, \tag{40}$$

whose Gershgorin bounds place the ellipticity in $[0.51, 1.20]$. The manufactured solution is $u = 0.25 + 0.10x - 0.08y + 0.06z + \sin \pi x \, \sin \pi y \, \sin \pi z \, \big(1 + 0.20 \sin \pi(x+y) + 0.15 \cos \pi(y+z)\big)$, with forcing obtained by automatic differentiation and Dirichlet data on all six faces. The frozen network follows the protocol of Appendix C.2, with $4{,}000$ interior and 400 boundary training points and the budget calibrated by the same procedure to $5{,}000$ steps. Observations are 500 interior entries placed uniformly at random in the cube plus 50 per face, so at order 800 the dense matrix and its Cholesky factor require roughly 10 MB together in float64, about 5 MB each; the error is the unweighted relative discrete L^2 error of Appendix C.1 on a 24^3 uniform tensor grid.

Table 27 reports the outcome. The correction reduces the relative error from $(8.15 \pm 1.56) \times 10^{-3}$ to $(4.18 \pm 0.45) \times 10^{-4}$, a geometric-mean improvement of $19.29\times$, with error-field correlation 0.998131 ± 0.000774. It takes 25.1 ± 5.1 s against 38.1 ± 1.3 s of network training, 65.7% of the training cost. This single smooth, constant-coefficient case demonstrates the construction in three dimensions at 800 observations. Its computational cost is substantial relative to training; comparisons with alternative solvers and growth in observation count remain open.

Table 27: Smooth anisotropic diffusion on the unit cube, a single three-dimensional feasibility case: mean $\pm$ sample standard deviation over seven independent runs. No matched comparator was run on this benchmark.

Quantity	Value
Raw PINN relative L^2	$(8.15 \pm 1.56) \times 10^{-3}$
Corrected relative L^2	$(4.18 \pm 0.45) \times 10^{-4}$
Mean $\log_{10}$ reduction	1.285 ± 0.089
Geometric-mean improvement	19.29
Error-field correlation	0.998131 ± 0.000774
Correction wall-clock (s)	25.1 ± 5.1
Network training wall-clock (s)	38.1 ± 1.3
Correction cost relative to training	65.7%
Matched comparator	none run

H Error Coordinates, Solution Coordinates, and the Direct Kernel Solver

Neural-operator surrogates learn maps between PDE inputs and solution fields (Kovachki et al., 2023). DeepONet uses branch–trunk representations (Lu et al., 2021), and Fourier neural operators parameterize kernel layers in Fourier space (Li et al., 2021); PINO additionally incorporates PDE constraints during operator learning (Li et al., 2024). In the present comparison the supplied field is a PINN for one fixed instance, and the direct kernel solver reconstructs that instance from its governing data.

Affine equivalence. For a linear problem let d collect the PDE and boundary data. With the same kernel and the same observation covariance Σ, the auxiliary error-coordinate model $e \sim \mathcal{GP}(0, k)$ with observations $\Lambda e = d - \Lambda u_\theta$ and the solution-coordinate model $u \sim \mathcal{GP}(u_\theta, k)$ with observations $\Lambda u = d$ give identical conditional means and covariances under $u = u_\theta + e$. In the linear case the error-coordinate construction is therefore an exact reformulation of kernel solution with prior mean u_θ: the frozen network supplies the defect, and the operator L does not depend on it.

For a fixed linearization point the same equivalence holds in the nonlinear case. With $A_\theta = N'(u_\theta)$ frozen, the error-coordinate model for the Newton correction δ, with observations $\Lambda\delta = y$, and the solution-coordinate model $w \sim \mathcal{GP}(u_\theta, k)$, with observations $A_\theta w = A_\theta u_\theta - r$ at the interior sites and $B_j w = g_j$ on the faces, are affine reparameterizations of each other under $w = u_\theta + \delta$. The dependence of the nonlinear correction on the frozen PINN therefore does not come from the choice of coordinates. It enters through the starting field u_θ, its residual r and the operator $N'(u_\theta)$ around which the step is constructed. A zero-mean solver for the original nonlinear PDE poses a different problem, since it must treat the nonlinearity itself, and is not a reparameterization of this frozen linearized step.

A zero-mean direct solver on the same sites. A different model places a zero-mean prior directly on the solution and conditions on the original PDE data at the same sites, with independently fitted hyperparameters, so it differs from the correction in both its prior mean and its hyperparameters. Table 28 compares it with the post-hoc correction on the seven linear benchmarks over seven independent runs; the two canonical nonlinear benchmarks are not applicable, because this direct solver is defined for linear problems. Both models use the same observation sites, kernel family, optimizer starts, bounds, profiled amplitude and jitter ladder (Appendix B.6), and each is fitted independently. Their recorded wall-clock, covering hyperparameter fitting, conditioning and field readout and excluding the cost of producing the frozen network, is comparable: the per-benchmark means over seven runs lie between 6.5 and 14.6 s for the direct solver and between 6.8 and 14.0 s for the post-hoc correction. The two solvers were timed side by side. A separate session on the same hardware produced the canonical correction timings of 7.7 to 16.4 s on the linear benchmarks in Table 7 (Appendix B.10). The corrected errors themselves are the same recorded outputs in both tables and agree to every digit printed.

The paired geometric-mean error ratio of the post-hoc correction to the direct solver is below one on Poisson 1D (0.0717) and reaction–diffusion (0.287), and above one on heat (1.42), wave (4.00), convection (130), Poisson 2D (664) and Helmholtz 2D (1.33×10^3). These are point estimates over seven runs; by the interval estimates of Table 10 the direct solver is resolved as more accurate on four benchmarks — wave, convection, Poisson 2D and Helmholtz 2D — the correction is resolved as more accurate on Poisson 1D and reaction–diffusion, and heat is undetermined. This comparison measures the joint effect of prior mean and independently fitted hyperparameters. The affine equivalence above holds when the kernel and regularization are shared.

I The Strengthened Neural Error-Field Control

This appendix specifies the scaled neural control and the complete-cost comparison, fixed in full before the experiments were run. The neural error-field control uses the study-specific implementation detailed below.

Table 28: Post-hoc correction versus a zero-mean direct kernel solver on the same observation sites: final relative L^2 error, mean $\pm$ sample standard deviation over seven independent runs, and the paired geometric-mean error ratio (post-hoc / direct) with the standard deviation of its $\log_{10}$. †Ratio above one, where the direct solver is more accurate. The direct solver is defined for linear problems only.

Benchmark	Post-hoc correction	Direct zero-mean solver	Ours / direct ratio	s.d. $\log_{10}$ ratio
Poisson 1D	$(4.84 \pm 5.17) \times 10^{-10}$	$(4.03 \pm 1.25) \times 10^{-9}$	0.0717	0.51
Reaction–diffusion 1D	$(1.96 \pm 0.58) \times 10^{-8}$	$(6.47 \pm 0.59) \times 10^{-8}$	0.287	0.17
Heat 1D	$(1.22 \pm 0.86) \times 10^{-7}$	$(7.03 \pm 1.19) \times 10^{-8}$	$1.42^\dagger$	0.25
Wave 1D	$(3.92 \pm 2.57) \times 10^{-6}$	$(8.06 \pm 1.96) \times 10^{-7}$	$4.00^\dagger$	0.37
Poisson 2D	$(8.54 \pm 2.79) \times 10^{-4}$	$(1.27 \pm 0.40) \times 10^{-6}$	$664^\dagger$	0.17
Helmholtz 2D	$(4.14 \pm 1.54) \times 10^{-3}$	$(3.13 \pm 1.18) \times 10^{-6}$	$1.33 \times 10^{3\dagger}$	0.21
Convection 1D	$(3.84 \pm 2.46) \times 10^{-3}$	$(8.62 \pm 18.48) \times 10^{-5}$	$130^\dagger$	0.63
Allen–Cahn 1D	*not applicable: the direct solver is defined for linear problems*			
Burgers 1D	*not applicable: the direct solver is defined for linear problems*			

Shared objective and architecture. The objective follows Table 5. At the same 500 interior and 200 condition sites, with the same frozen linearized operator and observation vector y, it minimizes the mean squared misfit of $A_\theta\varphi$ to y at the interior sites plus 100 times the weighted misfit at the condition sites. It retains the architecture $(d, 64, 64, 64, 64, 1)$ with tanh, the three restarts advanced round-robin in blocks of 100 steps, the checkpoint every block, the Adam constants and the schedule 10^{-3}, then 3×10^{-4} from step 1,000 and 10^{-4} from step 5,000, and selection by minimum observable operator-data loss.

Scaling, initialization and refinement. The control makes three adjustments, each using the known operator, residual and condition defects.

An operator-matched output scale. The original control has no explicit output scale. We test one operator-based normalization: the trained field is $\varphi = s\widehat{\varphi}$ with $s = \mathrm{rms}(y_{\mathrm{int}})/\mathrm{rms}(A_\theta\widehat{\varphi}_0)$, measured once per restart on that restart's Glorot draw (Glorot & Bengio, 2010). This matches an initial operator-action scale to the observed forcing scale; it is a conditioning heuristic based on the observed forcing and the initial network's operator action.

A zero output layer at initialization. The final weight matrix and bias start at zero, so the trajectory begins exactly at the uncorrected network and the initial loss is the observable norm of the data. The hidden layers keep their per-restart draw and the scale is measured before the output layer is zeroed.

A second-order refinement. Adam runs on $[0, 2T_i)$ and L-BFGS-B on $[2T_i, 4T_i)$ from the best Adam checkpoint, in blocks of at most 50 iterations with $\texttt{ftol} = 10^{-12}$ and $\texttt{gtol} = 10^{-8}$, the tolerances the operator-kernel fit itself uses, repeated until the deadline or until a block reports no progress. The switch at half the budget was fixed in advance.

Budget accounting. One trajectory per network with total budget $4T_i$, read at T_i and $4T_i$. A readout is the best checkpoint available before its deadline, with availability times stored alongside the measurements. The T_i readout is pure Adam; the $4T_i$ readout includes the refinement. These are two points on one fixed $4T_i$ trajectory, with a common fixed recipe and paired readouts. The control is handed the observation vector and charged the full T_i, which includes the cost the correction pays to build it, so the matching favours the control.

The new timing bracket includes observation construction, kernel fitting, Stage 2 where applicable, field readout and the accompanying quantity-of-interest certificate calculations. It is measured with device synchronization in a warmed process, with compilation charged to each arm. The historical bracket used for Table 1 starts after observation construction. The correction is recomputed and timed in the full-cost cohort: all 28 raw-network errors reproduce exactly, and the largest relative change in corrected error is 1.41×10^{-4}, within the prespecified 10^{-3} tolerance. The full-cost comparisons use these paired errors and times.

All four benchmarks retain all seven original frozen networks, seeds 0 to 6, giving 28 completed units. Two earlier launches were aborted after accounting and compilation defects were identified; their partial records are archived separately. The reported results use the complete rerun with corrected budget accounting and compilation. Blocks that finish after the final deadline can increase elapsed job time beyond $4T_i$; the reported budget and errors use checkpoints available by the deadline.

The tested problems mix second-order elliptic, first-order transport and nonlinear operators with variable coefficients and nonzero boundary defects. All use the same initialization and operator-based normalization specified above.

Table 29: Error against additional wall-clock on the four re-measured benchmarks, one timing cohort, mean ± sample standard deviation over seven runs. The additional time is the budget each arm is charged; base PINN training is a shared sunk cost and is excluded.

Benchmark	Arm	Budget	Additional time (s)	Relative L^2
Allen–Cahn 1D	operator-kernel	1×	59.3	$(2.04 \pm 1.41) \times 10^{-3}$
	strengthened neural	1×	59.3	$(1.06 \pm 0.42) \times 10^{-2}$
	strengthened neural	4×	237.1	$(1.56 \pm 0.82) \times 10^{-3}$
	continued training	1×	59.3	$(8.08 \pm 1.87) \times 10^{-3}$
	continued training	4×	237.1	$(6.15 \pm 2.62) \times 10^{-3}$
Burgers 1D	operator-kernel	1×	67.8	$(2.24 \pm 1.20) \times 10^{-3}$
	strengthened neural	1×	67.8	$(2.89 \pm 2.77) \times 10^{-3}$
	strengthened neural	4×	271.0	$(1.14 \pm 2.51) \times 10^{-3}$
	continued training	1×	67.8	$(5.02 \pm 1.79) \times 10^{-4}$
	continued training	4×	271.0	$(2.15 \pm 1.29) \times 10^{-4}$
Convection 1D	operator-kernel	1×	8.6	$(3.84 \pm 2.46) \times 10^{-3}$
	strengthened neural	1×	8.6	$(3.27 \pm 1.10) \times 10^{-3}$
	strengthened neural	4×	34.5	$(2.18 \pm 0.39) \times 10^{-3}$
	continued training	1×	8.6	$(3.52 \pm 2.57) \times 10^{-3}$
	continued training	4×	34.5	$(3.11 \pm 2.62) \times 10^{-3}$
Helmholtz 2D	operator-kernel	1×	13.8	$(4.14 \pm 1.54) \times 10^{-3}$
	strengthened neural	1×	13.8	$(2.07 \pm 1.38) \times 10^{-2}$
	strengthened neural	4×	55.0	$(1.06 \pm 0.36) \times 10^{-2}$
	continued training	1×	13.8	$(1.01 \pm 0.56) \times 10^{-2}$
	continued training	4×	55.0	$(6.50 \pm 2.85) \times 10^{-3}$

Comparison with the original control. The geometric-mean raw-to-control improvements of the new T_i readout and the original control, respectively, are 1.263 and 1.113 on Allen–Cahn, 3.293 and 0.988 on Burgers, 2.535 and 1.523 on convection, and 0.971 and 1.248 on Helmholtz 2D. Both columns use geometric means. These controls share the frozen starting networks but come from different executions and timing brackets, so recipe changes and timing scope vary together. On Helmholtz, the original correction-to-control paired error ratio is 0.294 with interval $[0.180, 0.480]$, while the new-cohort ratio at T_i is 0.228 with interval $[0.151, 0.346]$; each interval belongs to its own cohort.

Output-scale diagnostics. Averaged over runs and restarts, $\mathrm{rms}(A_\theta \widehat{\varphi}_0)$ is 6.32×10^{-1}, 1.80×10^{-1} and 2.54 on Allen–Cahn, Burgers and convection but 6.30×10^{-2} on Helmholtz, while $\mathrm{rms}(y_{\mathrm{int}})$ is 9.39×10^{-1} there against 2.21×10^{-2}, 8.49×10^{-3} and 1.92×10^{-2}. The resulting mean scale is 21.8 on Helmholtz and between 0.011 and 0.163 on the other three. The normalization therefore assigns a much larger scale on Helmholtz. Weak operator action on the random initial field is a possible explanation; separating the Laplacian and zeroth-order contributions would be needed to test it. The canonical $k = 1$ Dirichlet operator is negative definite (Appendix C.1). The T_i readout follows $3{,}043$ Adam steps under the fixed rule, making the observed comparison specific to this initialization, optimizer schedule and deadline.

Paired accuracy comparisons. At matched cost, the correction-to-scaled-control error ratios are 0.146 ($[0.071, 0.300]$) on Allen–Cahn and 0.228 ($[0.151, 0.346]$) on Helmholtz 2D (Table 29). The ratios on Burgers, 1.11 ($[0.415, 2.96]$), and convection, 1.07 ($[0.526, 2.19]$), leave the direction unresolved. At $4T_i$, the neural control is more accurate on Burgers, with ratio 6.94 ($[1.49, 32.3]$), while the correction remains more accurate on Helmholtz, 0.386 ($[0.304, 0.491]$). Allen–Cahn, 1.02 ($[0.386, 2.71]$), and convection, 1.55 ($[0.861, 2.81]$), are unresolved. Continued training improves on the correction on Burgers at T_i by $4.33\times$ ($[2.82, 6.65]$), rising to $10.7\times$ ($[5.41, 21.0]$) at $4T_i$.

J A Nonlinear Variable-Coefficient Problem with Direct Forcing and a Numerical Reference

Every canonical benchmark has an analytically known solution, fixed before anything is computed: five of the nine are homogeneous problems with a classical exact solution and zero forcing, and the other four are manufactured, with the forcing defined as the operator applied to the chosen solution. This appendix adds one problem specified through its coefficients and forcing, with no prescribed analytic solution. On $\Omega = (0, 1)^2$ with homogeneous Dirichlet data on the whole boundary,

$$-\nabla\cdot\big(a(x, y)\,\nabla u\big) + 5u^3 = f(x, y), \tag{41}$$

$$a(x, y) = 0.02\left[1 + 0.5\sin(2\pi x)\sin(2\pi y)\right] \in [0.01, 0.03], \tag{42}$$

$$f(x, y) = 2\exp\Big(-\tfrac{(x-0.30)^2+(y-0.35)^2}{0.12^2}\Big) - 1.5\exp\Big(-\tfrac{(x-0.70)^2+(y-0.65)^2}{0.10^2}\Big). \tag{43}$$

The forcing is prescribed directly rather than derived from a chosen u, and evaluation uses the independent numerical reference below. The coefficient is small and varies by a factor of three across the domain. The resolved solution has a positive lobe peaking at 0.524 near the first source and a negative lobe reaching -0.362 near the second, and its gradient magnitude varies by only a factor of 2.3 between its maximum and its mean, so the field is broad rather than sharply localized. The nonlinearity is not perturbative: at the peak $5u^3 = 0.72$ against a forcing amplitude of 2, so the reaction term balances a substantial part of the forcing. The linearized operator seen by the correction is $A_\theta v = -\nabla\cdot(a\nabla v) + 15u_\theta^2 v$, and the divergence form is implemented as $-a\Delta v - \nabla a\cdot\nabla v + 15u_\theta^2 v$; dropping the $\nabla a\cdot\nabla v$ term changes the operator by 5.1×10^{-2} in absolute value on a probe field, so the variable coefficient gradient contributes measurably to this probe. The automatic-differentiation operator agrees with an independently coded flux difference to 2.3×10^{-8} relative.

Base networks and correction. Seven networks, seeds 0 to 6, use the primary recipe of Appendix C.2 unchanged: 4×64 tanh, float64, $4{,}000$ interior and 400 boundary collocation points, boundary weight 100, Adam at 10^{-3}. The step budget is fixed at $40{,}000$ before any run, matching the value Poisson 2D uses, independently of reference error. The correction follows Algorithm 1: 500 interior and 200 boundary sites under the same scrambled-Sobol mechanism, the squared-exponential kernel, the same fit, jitter and block-weighting rules, Stage 2 active, and a 200×200 evaluation grid. All seven runs are retained. This is a separate cohort and is not added to the nine-benchmark canonical aggregates.

The independent reference. The reference solves equation 41 by a conservative five-point finite-volume discretization on a uniform grid, with face coefficients evaluated exactly at the midpoints, boundary nodes eliminated, and damped Newton with backtracking on $\|F\|_2$ from $u^0 = 0$, one sparse LU factorization per iteration. The reference solver is implemented independently of the estimator. The grid ladder $65, 129, 257, 513$ per axis and the acceptance conditions were fixed before any comparison: the discrete normalized residual $\|F\|_\infty/\max(1, \|f\|_\infty)$ at most 10^{-10}; the relative L^2 difference between the 257 and 513 solutions, both interpolated to the same evaluation grid, at most 10^{-4}; and an observed order from the last two refinements inside $[1.5, 2.5]$.

All three conditions hold: the largest normalized residual is 6.2×10^{-11}, the 257-to-513 difference is 5.2×10^{-5}, and the observed order is 1.9935. The last difference is an empirical reference-resolution indicator, more than two orders of magnitude below the mean uncorrected error but about half the mean corrected error. It is not a certified bound to the continuum solution, and precise corrected-error values and small differences between correction variants can be sensitive to reference resolution. The full CPU reference ladder costs 8.75 s including assembly and solve; the 513 grid

Table 30: Reference ladder for equation 41. Newton converges in five iterations at every resolution. The observed order is $\log_2$ of the ratio of successive differences on the common evaluation grid. Timings are single measurements on one CPU core.

Points/axis	Unknowns	Newton	Norm. residual	Diff. from previous	Order
65	3,969	5	6.20×10^{-11}	—	—
129	16,129	5	6.13×10^{-11}	8.26×10^{-4}	—
257	65,025	5	6.14×10^{-11}	2.06×10^{-4}	2.0028
513	261,121	5	6.20×10^{-11}	5.17×10^{-5}	1.9935

accounts for $7.45\,\mathrm{s}$ and has $261{,}121$ unknowns. These are the measured costs of the reference implementation, with the correction timing scope specified in Appendix I.

The reference enters evaluation only. It is not read by the hyperparameter fit, the regularization, the checkpoint selection of any arm, or the training of the seven networks.

Sensitivity to the recorded reference grid. The two finest recorded reference fields also give a deterministic check of the large comparison margins, without retraining or recomputing a prediction. Let r_f, r_c be their interpolants on the evaluation grid, let $d_g = \|r_f - r_c\|_2/\|r_f\|_2 = 5.1721\times10^{-5}$, and write $E_f(v) = \|v - r_f\|_2/\|r_f\|_2$ for a fixed prediction v. The triangle inequality gives

$$\frac{\|v_o - r_c\|_2}{\|v_c - r_c\|_2} \le \frac{E_f(v_o) + d_g}{E_f(v_c) - d_g}, \qquad E_f(v_c) > d_g, \tag{44}$$

where v_o is the correction and v_c a comparator. Every upper bound is below one for both controls at both budgets on all seven seeds: all 28 pairwise advantages persist if the recorded 257-grid reference replaces the 513-grid reference. Similarly, the geometric mean of $(E_f(u_\theta) - d_g)/(E_f(v_o) + d_g)$ is 71.44, a lower bound on the raw-to-corrected improvement under that substitution. These deterministic norm bounds quantify sensitivity to the choice between the two recorded reference grids; continuum discretization error remains a separate quantity.

Stage 2 on this operator. The nonlinearity constant is estimated from operator probes on the first 64 interior sites. Across the seven runs, C_N is 7.24 on average, ranging over 6.69 to 7.85, and the mean relative nugget rises from 5.87×10^{-9} to 9.94×10^{-3}. The single fitted stage reaches a geometric-mean improvement of $152\times$ against the two-stage method's $115\times$; paired within runs the ratio (Stage 2 / Stage 1) is 1.32 with interval $[0.996, 1.74]$. Its point estimate favours Stage 1, but the interval includes one and does not resolve an accuracy difference. The Stage 1 final relative error is $(1.01 \pm 1.28)\times10^{-4}$. Across the three nonlinear problems, the component ablation resolves an increase in error from Stage 2 on Allen–Cahn and leaves the differences on Burgers and this problem unresolved (Appendix E.3). The main comparisons use the complete two-stage method.

Table 31: Additional-compute arms on the new problem, mean $\pm$ sample standard deviation over seven runs. The additional time is the budget each arm is charged; base training is a shared sunk cost and is excluded.

Benchmark	Arm	Budget	Additional time (s)	Relative L^2
Nonlinear var.-coeff. 2D	operator-kernel	$1\times$	81.0	$(1.08\pm0.85)\times10^{-4}$
	strengthened neural	$1\times$	81.0	$(7.44\pm2.24)\times10^{-3}$
	strengthened neural	$4\times$	324.1	$(3.06\pm2.38)\times10^{-3}$
	continued training	$1\times$	81.0	$(3.22\pm0.73)\times10^{-3}$
	continued training	$4\times$	324.1	$(1.60\pm0.31)\times10^{-3}$

These measurements cover one physical instance with seven training repetitions.