BEYOND SIMULATION: RETAIN-AND-REPAIR NEURAL OPERATORS FOR REAL-WORLD ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators increasingly benefit from pretraining on numerical simulations, yet adapting them for real-world prediction remains challenging. We introduce the Retain-and-Repair Neural Operator (R^2NO), a framework for adapting simulation-pretrained operators to real-world data while retaining useful pretrained structure. The pretrained operator is first finetuned on real data and then frozen to provide a source prediction, and a shared repair module learns a sequence of refinements from the same observations. Using orthogonal Fourier projections, a spectral ensemble fits a small ridge regression within each cell of the Fourier domain and combines the refinements by weights fitted on a held-out split of the real data. The cells are defined jointly by radial ranges, angular sectors, and measured channels, allowing refinement depth to vary with frequency magnitude, with orientation, and across channels. Including the source prediction as a candidate makes retention available in every cell, and independently trained repair modules enter the same combination as additional candidates. On all RealPDEBench systems and six backbones, R^2NO consistently outperforms full finetuning and iterative refinement. The framework treats adaptation depth as a cell-specific choice learned from real data.

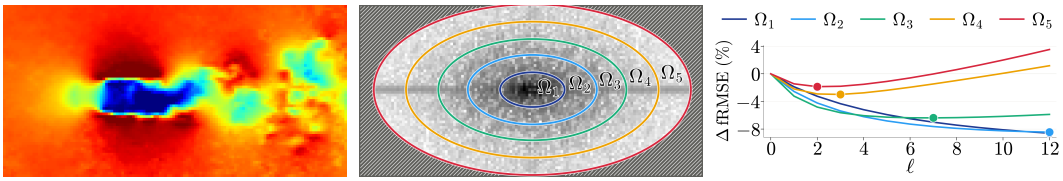
1 INTRODUCTION

Neural operators learn mappings between function spaces and provide fast surrogates for the evolution of physical systems (Li et al., 2020; Lu et al., 2021; Raonic et al., 2023). Much of their development has relied on numerical simulations for both training and evaluation, while recent pretraining approaches have learned reusable representations across diverse simulated systems (Hao et al., 2024; Herde et al., 2024). Even when simulations and real measurements nominally follow the same governing equations, modeling approximations, experimental conditions, and measurement processes can create a substantial sim-to-real gap. However, adapting neural operators to limited real measurements remains largely underexplored. This motivates a central question: how can we adapt simulation-trained operators to real measurements without discarding what they already get right?

Looped models provide a natural structure for addressing this question. Recent approaches in language and diffusion modeling repeatedly apply a weight-shared block to produce intermediate representations without increasing the number of parameters (Giannou et al., 2023; Saunshi et al., 2025; Movahedi et al., 2026). In neural operators, IRNO (Liu et al., 2026) similarly applies iterative corrections to mitigate spectral bias. We use this iterative structure for a different purpose: determining which components of a simulation-pretrained prediction should be retained and which should be repaired using real observations.

We analyze this behavior in detail in Section 2, showing that different spectral regions along a fixed refinement trajectory favor different depths. Consequently, a single iterate trades accuracy in one region against accuracy in another. The Fourier domain is particularly suitable for resolving this tradeoff because disjoint spectral regions induce mutually orthogonal projections, allowing the squared prediction error to be decomposed exactly across regions. Motivated by this observation, we introduce the Retain-and-Repair Neural Operator (R^2NO), a model-agnostic framework for adapting simulation-pretrained neural operators to real measurements. R^2NO adapts in two phases: it first finetunes the pretrained operator on real observations, then freezes the result and applies a shared repair module trained on the same observations, producing a trajectory of candidate predictions. A spectral ensemble then combines these candidates separately within Fourier cells defined by radial

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.



(a) Real observation (b) Radial Fourier regions (c) Refinement error by region

Figure 1: **Different Fourier regions favor different refinement depths.** (a) A real Cylinder observation in physical space. (b) Five radial Fourier regions, ordered from low to high frequency. (c) Region-wise fRMSE change (%) relative to $\ell = 0$ along an IRNO trajectory trained on FNO; markers indicate minima over the evaluated depths, and lower is better. Setup in Appendix C.1.

frequency, orientation, and channel. Because the original prediction remains an explicit candidate, each cell can retain transferable simulation-derived components while using the refinement trajectory to repair components that do not transfer. The cell-wise weights are fitted through small ridge-regression problems on real data, and independently trained repair trajectories can be incorporated as additional candidates. Since R^{2}NO operates on predicted fields, it can be applied across neural operator backbones without modifying their internal architectures.

We make the following contributions.

- **Model-agnostic retain-and-repair adaptation.** We introduce a framework that adapts a simulation-pretrained neural operator using a weight-shared repair module trained on limited real observations, while preserving the source prediction as an explicit candidate.
- **Region-wise spectral ensemble and analysis.** We fit region-specific combinations of the source prediction and its repairs over a radial–angular spectral partition. We characterize the advantage over using a common refinement depth and distinguish convergence of the repair trajectory from the accuracy of its finite iterates.
- **Evaluation under real-world finetuning.** We evaluate R^{2}NO on all five RealPDEBench systems (Hu et al., 2026) and six pretrained backbones, comparing it with full finetuning and iterative refinement under the same sim-to-real adaptation setting.

2 Motivation: Frequency-Dependent Refinement

A shared refinement module produces several predictions for the same input, but how many times to apply it is normally decided once for the whole output field. Methods that adapt the depth do so per token (Elbayad et al., 2019; Bae et al., 2026), not across the frequency content of a field. We ask whether every spectral component of a real-world prediction favors the same depth.

Iterative residual refinement. The iterative refinement neural operator (IRNO) of Liu et al. (2026) augments a pretrained neural operator with a weight-shared residual loop, so one trained model yields a prediction at every depth. The operator $\Psi(\cdot;\pi)$ stays frozen and maps a real input history $\mathcal{X}$ to an initial prediction $h_0 = \Psi(\mathcal{X};\pi)$. A refinement module $\Phi(\cdot;\theta)$ receives the input history together with the current prediction and returns a residual of the same shape as the output field, which is added with a step size $\alpha > 0$.

$$h_{\ell+1} = h_\ell + \alpha\,\Phi(\mathcal{X}, h_\ell;\theta), \qquad \ell = 0,\dots,L-1. \tag{1}$$

Because Φ predicts a correction rather than the field itself, the pretrained prediction is the starting point of the loop and every iterate h_ℓ is a complete prediction, while running the loop longer adds computation but no parameters. IRNO trains Φ with an objective that supervises every iterate against the target, and at inference it reports the field at one depth chosen for the whole output.

Refinement trained on real data. For a RealPDEBench system, we take the finetuned backbone as Ψ, train Φ on the real training trajectories with the IRNO objective, and then hold both π and θ fixed. Rather than scoring the whole field at one depth, we look at how the error of each iterate is distributed over the Fourier domain.

Error by spectral region. Let $\mathcal{F}$ be the unitary Fourier transform over the spatial axes of the output, and let $\Omega_1, \ldots, \Omega_{N_\Omega}$ be disjoint, conjugate-symmetric sets of wavenumbers. Region b acts on a field by keeping only its Fourier coefficients in Ω_b.

$$P_b v = \mathcal{F}^{-1}[\mathbf{1}_{\Omega_b} \odot \mathcal{F} v], \qquad b = 1, \ldots, N_\Omega. \tag{2}$$

The projections in Eq. 2 are mutually orthogonal, since $\mathcal{F}$ is unitary and the sets Ω_b are disjoint, and conjugate symmetry keeps $P_b v$ real for real v. We run the loop of Eq. 1 for $L = 12$ steps on the real test split $\mathcal{D}^{\text{test}}$ and compute the fRMSE (Takamoto et al., 2022; Hu et al., 2026) of every iterate $h_0, \ldots, h_L$ in the $N_\Omega = 5$ radial regions of Figure 1(b); Appendix C.1 details the setup. Figure 1(c) plots the change relative to $\ell = 0$; negative values mean improvement, and the markers indicate each region's best depth. On the real Cylinder data, this best depth decreases from the lowest band Ω_1 to the highest Ω_5: no single ℓ serves the whole spectrum. This is an observation about one trained trajectory rather than a rule tied to frequency; what it argues for is learning each region's depth from data instead of prescribing it.

The cost of a common depth. The disagreement can be quantified in the squared field error that our spectral ensemble later minimizes. Let $\widehat{\mathbb{E}}_{\text{test}}$ denote the empirical average over the input–target pairs $(\mathcal{X}, y)$ of $\mathcal{D}^{\text{test}}$, let all norms be Euclidean norms of the vectorized field, and let $\mathcal{S}_0 = \{0, \ldots, L\}$ be the set of candidate depths, which includes the source prediction at $\ell = 0$. For every candidate we record its squared error restricted to region b.

$$\widehat{E}_b(\ell) = \widehat{\mathbb{E}}_{\text{test}} \, \|P_b(h_\ell - y)\|_2^2, \qquad \ell \in \mathcal{S}_0. \tag{3}$$

When the regions form an exhaustive partition of the represented wavenumbers, every field is the sum of its regional components, and orthogonality splits its squared error exactly across regions.

$$\sum_{b=1}^{N_\Omega} P_b v = v, \qquad \|v - y\|_2^2 = \sum_{b=1}^{N_\Omega} \|P_b(v - y)\|_2^2. \tag{4}$$

Averaged over $\mathcal{D}^{\text{test}}$, the error of candidate h_ℓ is therefore $\sum_b \widehat{E}_b(\ell)$, and the price of one depth on every region is the gap between the best common depth and independent region-wise selection.

$$\min_{\ell \in \mathcal{S}_0} \sum_{b=1}^{N_\Omega} \widehat{E}_b(\ell) - \sum_{b=1}^{N_\Omega} \min_{\ell \in \mathcal{S}_0} \widehat{E}_b(\ell) = \min_{\ell \in \mathcal{S}_0} \sum_{b=1}^{N_\Omega} \Big[\widehat{E}_b(\ell) - \min_{\ell' \in \mathcal{S}_0} \widehat{E}_b(\ell') \Big] \geq 0. \tag{5}$$

Each bracket is nonnegative, so the gap is strictly positive exactly when no single depth minimizes every regional error, and the subtracted term is attained by the composite field $\sum_b P_b h_{\ell_b}$, with ℓ_b a minimizer of $\widehat{E}_b$, since Eq. 4 applies to that field as well. The five radial bands serve only to expose the frequency dependence. The spectral ensemble of Section 3.3 applies the same principle to an exhaustive partition of the represented Fourier grid, replaces the selection of one candidate per band by a fitted combination of all of them, fits that combination on a split of the real data disjoint from the one used to train the repair module, and keeps the source prediction among the candidates so that any cell may retain it exactly.

3 Method: Retain-and-Repair Neural Operators

Motivated by Section 2, R^{2}NO does not commit to one refinement depth. It keeps the whole trajectory of the repair loop, the source prediction included, and lets real data decide, cell by cell in the Fourier domain, how much of each to use.

3.1 Overview

Figure 2 shows the two stages. Section 3.2 describes the retain-and-repair loop, which runs the refinement module of Eq. 1 from the frozen source prediction and keeps its iterates as candidates, and Section 3.3 describes the spectral ensemble, which fits, for each Fourier cell, how the output draws on those candidates and on the source prediction. The two stages use disjoint real data. The repair modules are trained on the training split $\mathcal{D}^{\text{train}}$ of the benchmark, and the ensemble is fitted on its validation split $\mathcal{D}^{\text{fit}}$, on whose targets the modules are never trained. Fitting on the training split

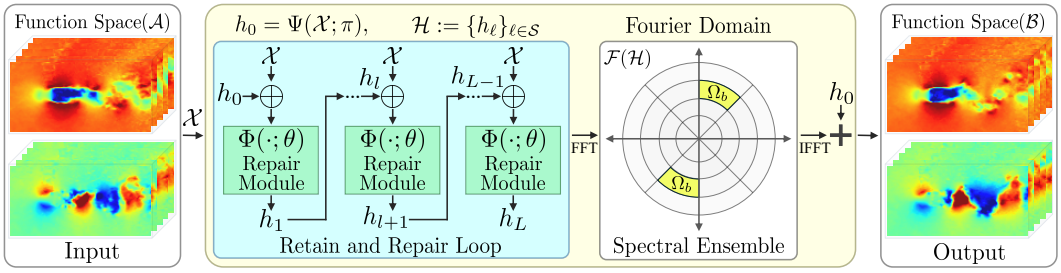


Figure 2: **Overall pipeline of R^{2}NO.** The retain-and-repair loop applies the shared repair module L times to the frozen source prediction h_0 and keeps every iterate. The spectral ensemble then fits, for each Fourier cell, how much of the source prediction and of each repair to use.

instead would be biased, because the repairs look better on their own training data than on new data and an ensemble fitted there would over-trust them relative to the source prediction. The validation split also selects the checkpoint of each repair module among those saved during training, and Appendix B accounts for this choice. The test split $\mathcal{D}^{\text{test}}$ is used only for evaluation. The pretrained neural operator $\Psi(\cdot;\pi)$ is first finetuned on $\mathcal{D}^{\text{train}}$ and then frozen, and it maps a real input history $\mathcal{X} \in \mathcal{A}$ to the source prediction $h_0 = \Psi(\mathcal{X};\pi) \in \mathcal{B}$, where $\mathcal{A}$ and $\mathcal{B}$ are the input and output function spaces in Figure 2. This h_0 coincides with the prediction of the full finetuning baseline, although any other checkpoint could serve as the source. Appendix F evaluates the simulation-pretrained checkpoint as the source. Since the benchmark's systems report different variables at different resolutions, the modality and resolution mappings it prescribes are applied to the source prediction first, so that h_0 and the real target y share the same variables, forecast frames, and spatial grid. All residuals and Fourier cells below are defined on this common grid.

3.2 Retain-and-Repair Loop on Real Data

The repair module $\Phi(\cdot;\theta)$ is applied L times, starting from the source prediction h_0 with a step size $\alpha > 0$, and every iterate is kept as a candidate. When M modules are trained independently from different random seeds, member m produces its own trajectory with parameters $\theta^{(m)}$, and the superscript is dropped for $M = 1$.

$$h_{\ell+1}^{(m)} = h_\ell^{(m)} + \alpha\,\Phi\Big(\mathcal{X}, h_\ell^{(m)}; \theta^{(m)}\Big), \qquad h_0^{(m)} = h_0, \qquad \ell = 0, \dots, L-1. \tag{6}$$

The source prediction enters Eq. 6 only as its starting point. Retention is enforced later, at the ensemble stage, through the residual form of Eq. 11, so the learned updates remain free to change any part of the field. In Figure 2, $\oplus$ denotes the concatenation of the module's inputs.

Training objective. Each module is trained on $\mathcal{D}^{\text{train}}$ with the objective and optimization procedure of the iterative refinement baseline (Liu et al., 2026), so that any difference in accuracy is attributable to the ensemble rather than to a modified training signal. The objective supervises every iterate of the loop with three terms: a spatial residual, a spectral residual, and a fixed-point penalty. All three are computed on the channels the benchmark measures (two of the three predicted channels for the fluid systems, one of sixteen for combustion), and in this paragraph $\|v\|^2$ denotes the mean of v^2 over those channels, the forecast frames, and the spatial grid, a mean rather than the Euclidean sum of Section 2, so that the terms have comparable scale.

The spatial residual of iterate ℓ is $\|h_\ell^{(m)} - y\|^2$. The spectral residual compares the magnitudes of the Fourier coefficients of the iterate and of the target, so it constrains the amplitude spectrum and leaves the phase to the spatial residual. It is averaged over the half-plane $\mathcal{K}_+$ of wavenumbers returned by the real-input Fourier transform, the other half being redundant for real fields, and over channels and frames as above, under a weighting μ_ℓ that increases with the radius $\rho(k) \in [0, \sqrt{2}]$ of the wavenumber normalized per axis by the Nyquist wavenumber:

$$\mathcal{L}_{\text{spe}}^{\ell} = \frac{1}{|\mathcal{K}_+|} \sum_{k\in\mathcal{K}_+} \mu_\ell(k) \Big(\big|[\mathcal{F}h_\ell^{(m)}](k)\big| - \big|[\mathcal{F}y](k)\big| \Big)^2, \qquad \mu_\ell(k) \propto 1 + \rho(k)^{\eta_\ell}, \tag{7}$$

with μ_ℓ in Eq. 7 normalized to unit mean over $\mathcal{K}_+$. The exponent $\eta_\ell = 1 + \frac{\ell-1}{L-1}$ rises from one at the first iterate to two at the last, so later iterates are pushed harder on high wavenumbers. The additive unit keeps the lowest wavenumbers from being discarded, and the weight at the corner of the grid is $1 + 2^{\eta_\ell/2}$ times that at the zero wavenumber, between 2.41 and 3.00. The unit-mean normalization keeps the overall scale of the term independent of η_ℓ. The fixed-point penalty $\|\Phi(\mathcal{X}, y; \theta^{(m)})\|^2$ evaluates the module at the real target and penalizes the correction it would still apply there, so that the target is encouraged to be a fixed point of the repair map (Hsieh et al., 2019).

Writing $\widehat{\mathbb{E}}_{\text{train}}$ for the empirical average over $\mathcal{D}^{\text{train}}$, the objective averages the spatial and spectral residuals over the L iterates and adds the penalty, with $\beta_{\text{spe}}, \beta_{\text{fp}} > 0$ weighting the spectral residual and the fixed-point penalty against the spatial residual:

$$\mathcal{L}\Big(\theta^{(m)}\Big) = \widehat{\mathbb{E}}_{\text{train}}\left[\frac{1}{L}\sum_{\ell=1}^{L}\Big(\big\|h_\ell^{(m)} - y\big\|^2 + \beta_{\text{spe}}\,\mathcal{L}_{\text{spe}}^{\ell}\Big) + \beta_{\text{fp}}\,\big\|\Phi\Big(\mathcal{X}, y; \theta^{(m)}\Big)\big\|^2\right]. \tag{8}$$

Only $\theta^{(m)}$ is updated by minimizing Eq. 8; the pretrained operator remains fixed.

3.3 Spectral Ensemble Across Refinement Depths

Fourier cells. The regions of Section 2 are refined into cells that separate frequency magnitude, orientation, and channel. For a spatial wavenumber k, let $\rho(k)$ be the normalized radius of Section 3.2, and let $\varphi(k) \in [0, 1)$ be the orientation of k modulo a half-turn, expressed as a fraction of the half-turn. The spectrum is divided into N_ρ radial bands and N_φ angular sectors, and the band and the sector containing k are

$$b_\rho(k) = \min\big(\lfloor \rho(k)\, N_\rho \rfloor,\, N_\rho - 1\big), \qquad b_\varphi(k) = \min\big(\lfloor \varphi(k)\, N_\varphi \rfloor,\, N_\varphi - 1\big). \tag{9}$$

The bands have equal width in ρ and the sectors equal width in φ. The clipping in Eq. 9 assigns the coefficients beyond the inscribed ellipse of the grid, where $\rho(k) \geq 1$, to the outermost band. A cell consists of the coefficients of one measured channel that share a band and a sector, so there are $N_\Omega = N_\rho N_\varphi N_{\text{ch}}$ cells for every system, in place of the five bands of Section 2. The projection P_b applies the mask of Eq. 2 to the channel of cell b and returns zero on the other channels. Since k and $-k$ share an orientation modulo a half-turn, a coefficient and its conjugate always fall in the same cell, which is why the two highlighted sectors in Figure 2 share one weight and every P_b maps real fields to real fields. The cells are disjoint, cover every coefficient of the measured channels, and are shared across forecast frames.

Candidate columns. Let $\mathcal{S} \subseteq \mathcal{S}_0 \setminus \{0\}$ be the retained depths and $\mathcal{H} := \{h_\ell\}_{\ell \in \mathcal{S}}$ the retained iterates of a module, as drawn in Figure 2. The columns of the ensemble are the residuals of every retained iterate relative to the source prediction, together with one base column,

$$c_1, \ldots, c_J \in \big\{h_\ell^{(m)} - h_0 :\ m \leq M,\ \ell \in \mathcal{S}\big\} \cup \{-h_0\}, \qquad J = M|\mathcal{S}| + 1. \tag{10}$$

Without the base column, a cell could only form affine combinations of h_0 and the iterates, and the base column lets it also attenuate the source prediction where its content does not transfer. The repair budget is M times the largest retained depth in evaluations of Φ, since every shallower depth is produced on the way at no extra cost.

Predictor and closed-form fit. The predictor is a residual correction of the source prediction, in which every cell adds its own combination of the columns.

$$\mathcal{R}(\mathcal{X}) = h_0 + \sum_{b=1}^{N_\Omega}\sum_{j=1}^{J} w_{b,j}\, P_b c_j. \tag{11}$$

All-zero weights reproduce h_0 exactly, and since every cell lies in a measured channel, the unmeasured channels of the output are always those of h_0. The weights $w_b \in \mathbb{R}^J$ of each cell are chosen to minimize the squared error on the fitting split, compared on the measured channels, with $\widehat{\mathbb{E}}_{\text{fit}}$ the empirical average over $\mathcal{D}^{\text{fit}}$. By Eq. 4 and the mutual orthogonality of the P_b, this error decomposes into one term per cell, and each term involves only the weights of that cell.

$$\widehat{\mathbb{E}}_{\text{fit}}\big\|y - \mathcal{R}(\mathcal{X})\big\|_2^2 = \sum_{b=1}^{N_\Omega}\widehat{\mathbb{E}}_{\text{fit}}\Big\|P_b(y - h_0) - \sum_{j=1}^{J} w_{b,j}\, P_b c_j\Big\|_2^2. \tag{12}$$

The fit therefore decouples into N_Ω independent least-squares problems of dimension J, and each of them is solved in closed form. Successive iterates are nearly collinear, so a plain least-squares solution would be unstable, and a ridge penalty is added. Let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product of vectorized fields. The Gram matrix of cell b collects the inner products of the projected columns with one another, and the right-hand side collects their inner products with the projected target residual.

$$[\mathbf{G}_b]_{jj'} = \widehat{\mathbb{E}}_{\mathrm{fit}}\big\langle P_b c_j, P_b c_{j'} \big\rangle, \qquad [\mathbf{r}_b]_j = \widehat{\mathbb{E}}_{\mathrm{fit}}\big\langle P_b(y - h_0), P_b c_j \big\rangle. \tag{13}$$

The ridge term is scaled by the mean diagonal entry of $\mathbf{G}_b$ so that the regularization is comparable across cells, and the weights of cell b follow by solving a $J \times J$ linear system.

$$w_b = \Big(\mathbf{G}_b + \lambda \, \frac{\operatorname{tr} \mathbf{G}_b}{J} \, \mathbf{I}\Big)^{-1} \mathbf{r}_b. \tag{14}$$

Here $\mathbf{I}$ is the identity matrix and $\lambda > 0$ is the ridge parameter, chosen once by solving on one half of the fitting split and scoring on the other over a fixed logarithmic grid. Eq. 14 is then re-solved on the whole split, and the weights are held fixed afterwards. No gradient descent is involved. Because $\mathcal{F}$ is unitary, $\|P_b v\|_2 = \|\mathbf{1}_{\Omega_b} \odot \mathcal{F} v\|_2$, so in practice the Gram matrices are accumulated from the masked Fourier coefficients of one forward transform per field, and the output is assembled with a single inverse transform, as drawn in Figure 2. All fields enter the fit in physical, denormalized units.

Properties of the fitted ensemble. The weights w_b are unconstrained, so a cell can reproduce any single column with a one-hot w_b or interpolate between columns, and neighboring cells may make different choices without any prescribed relation between position in the spectrum and depth. Setting $N_\Omega = 1$ recovers a single global combination. Since Eq. 12 is minimized cell by cell over all of these choices, at $\lambda = 0$ the fitting error of $\mathcal{R}$ is no larger than that of the source prediction, of any single candidate $h_\ell^{(m)}$, or of any cell-wise selection among them, and when the source checkpoint is the finetuned baseline, no larger than that of the baseline either. This is the ensemble counterpart of the gap in Eq. 5, stated for combinations rather than selections, and Appendix B gives the proof. The guarantee concerns the fitting split. The ridge term trades a controlled part of it for stability, and because $\mathcal{D}^{\mathrm{fit}}$ is disjoint from the split on which the candidates were trained, Appendix B also bounds how a cell-wise selection generalizes beyond it.

Multiple repair modules. For $M > 1$ the modules share the frozen source checkpoint and differ only in their random seed, as in deep ensembles (Lakshminarayanan et al., 2017), and their iterates enter Eq. 10 as additional columns of one joint fit. Since uniform averaging is one admissible choice of w_b in Eq. 12, the joint fit is never worse on the fitting split than averaging M separately fitted ensembles, and it lets a cell take a different depth from each member. The two enlargements differ in cost: adding depths to $\mathcal{S}$ reuses iterates already produced and only enlarges the $J \times J$ systems, whereas each additional member costs a training run and a rollout at inference.

Finally, convergence of the repair loop and accuracy of its iterates are different properties. Under a local contraction assumption, the loop converges to a fixed point whose distance to the target is controlled by the correction the module would still apply at the target, and even then the best depth can differ across regions. Appendix B makes both statements precise. The ensemble therefore combines h_0 and its repairs without requiring $\mathcal{R}$ to be a fixed point of any repair map.

4 Experiments

In this section, we compare R^{2}NO with the baselines on RealPDEBench and then examine the contribution of each of its components through ablations.

4.1 Real-World Adaptation on RealPDEBench

We evaluate whether the spectral ensemble improves real-world prediction across physical systems and pretrained backbones, following the real-world finetuning task of RealPDEBench (Hu et al., 2026), in which simulation-pretrained operators are adapted on real training trajectories and evaluated on held-out real observations. The five systems are Cylinder, Controlled Cylinder, fluid–structure interaction (FSI), Foil, and Combustion, and we retain the data partitions, preprocessing, observed variables, conditioning information, input and output windows, and evaluation procedure of the

Table 1: Real-world adaptation on RealPDEBench. All adapted methods start from the same finetuned checkpoint within each backbone, IRNO is reported at its final depth and at the depth $\ell^\star$ selected on the fitting split, and M counts the repair modules of R^{2}NO. Lower is better, and the best value of each column within a backbone is set in bold, with ties marked jointly.

		Cylinder		Controlled Cylinder		FSI		Foil		Combustion	
Backbone	Adaptation	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)
U-Net	Full finetuning	6.315	9.731	0.786	0.924	0.841	0.714	0.936	0.825	2.134	2.487
	IRNO (ℓ=12)	5.982	9.198	0.788	0.929	0.876	0.739	0.935	0.820	2.159	2.526
	IRNO ($\ell^\star$)	5.970	9.172	0.778	0.912	0.838	0.700	0.922	0.819	2.133	2.489
	R^{2}NO (M=1)	3.510	5.181	**0.776**	0.907	**0.836**	**0.697**	0.911	0.797	2.121	2.466
	R^{2}NO (M=3)	**2.775**	**3.912**	**0.776**	**0.905**	**0.836**	0.698	**0.905**	**0.784**	**2.119**	**2.462**
DeepONet	Full finetuning	6.608	10.698	2.913	5.186	3.315	4.762	2.256	2.893	2.270	2.746
	IRNO (ℓ=12)	4.061	6.179	0.956	1.234	1.114	1.147	1.477	1.900	2.188	2.593
	IRNO ($\ell^\star$)	4.061	6.179	0.956	1.234	1.114	1.147	1.477	1.900	2.188	2.593
	R^{2}NO (M=1)	3.121	3.984	0.854	1.029	0.971	0.856	1.316	1.611	2.159	2.536
	R^{2}NO (M=3)	**2.703**	**3.315**	**0.831**	**0.981**	**0.917**	**0.775**	**1.131**	**1.321**	**2.123**	**2.457**
FNO	Full finetuning	5.449	8.661	0.943	1.077	1.266	1.162	1.199	1.241	2.246	2.726
	IRNO (ℓ=12)	4.920	7.984	0.818	0.956	1.038	1.010	1.157	1.200	2.261	2.716
	IRNO ($\ell^\star$)	4.913	7.972	0.818	0.956	1.037	1.010	1.130	1.122	2.239	2.707
	R^{2}NO (M=1)	3.912	5.665	0.813	0.947	1.019	0.970	1.081	1.055	2.207	2.671
	R^{2}NO (M=3)	**3.432**	**4.919**	**0.804**	**0.934**	**0.985**	**0.927**	**1.061**	**1.019**	**2.201**	**2.659**
CNO	Full finetuning	3.524	5.709	0.798	0.903	0.960	0.779	1.145	1.343	2.370	2.954
	IRNO (ℓ=12)	3.223	4.687	0.785	0.902	0.858	0.710	1.062	1.119	2.220	2.632
	IRNO ($\ell^\star$)	3.223	4.683	0.782	0.898	0.858	0.709	1.040	1.099	2.219	2.637
	R^{2}NO (M=1)	2.838	3.846	**0.781**	**0.896**	0.856	0.702	0.996	1.019	2.199	2.606
	R^{2}NO (M=3)	**2.525**	**3.220**	**0.781**	**0.896**	**0.848**	**0.693**	**0.979**	**0.953**	**2.164**	**2.534**
Transolver	Full finetuning	8.641	12.887	1.665	2.156	2.260	2.507	1.965	2.243	3.744	4.943
	IRNO (ℓ=12)	4.912	7.750	0.883	1.021	1.024	0.822	1.243	1.370	2.442	2.858
	IRNO ($\ell^\star$)	4.912	7.750	0.883	1.021	1.024	0.822	1.243	1.370	2.396	2.770
	R^{2}NO (M=1)	3.305	4.439	0.841	0.976	0.964	0.752	1.127	1.178	2.209	2.604
	R^{2}NO (M=3)	**2.421**	**3.011**	**0.819**	**0.934**	**0.948**	**0.731**	**1.073**	**1.095**	**2.179**	**2.542**
DPOT-S	Full finetuning	4.404	6.726	0.847	0.987	0.990	0.737	1.093	1.178	2.106	2.408
	IRNO (ℓ=12)	4.151	6.399	0.799	0.943	0.869	0.733	1.057	1.155	2.108	2.350
	IRNO ($\ell^\star$)	4.151	6.399	0.799	0.943	0.869	0.733	1.057	1.155	2.082	2.343
	R^{2}NO (M=1)	3.224	4.990	0.795	0.931	0.866	0.729	1.037	1.140	2.064	2.327
	R^{2}NO (M=3)	**2.617**	**3.665**	**0.789**	**0.914**	**0.860**	**0.726**	**1.016**	**1.102**	**2.058**	**2.318**

benchmark. Its training split serves as $\mathcal{D}^{\text{train}}$ for the repair modules, its validation split serves as the fitting split $\mathcal{D}^{\text{fit}}$ on which the spectral ensemble and its ridge parameter are solved, and its test split is used only to compute the reported errors. Appendix C records the protocol in full.

Backbones and adaptation methods. We evaluate six backbones, namely U-Net (Ronneberger et al., 2015), DeepONet (Lu et al., 2021), CNO (Raonic et al., 2023), FNO (Li et al., 2020), Transolver (Wu et al., 2024), and DPOT-S (Hao et al., 2024), and Table 1 compares five adaptation settings for each of them. Full finetuning takes a backbone pretrained on simulation data and updates all its parameters on the real training split, and the resulting checkpoint serves as the frozen source operator of every other setting, so its prediction is exactly the source prediction h_0. IRNO (Liu et al., 2026) trains a shared refinement module on top of this checkpoint with the objective of Eq. 8, and we report it at its final iterate h_{12} and at the iterate $h_{\ell^\star}$ whose depth $\ell^\star \in \{0, \dots, 12\}$ minimizes the RMSE on the fitting split, which realizes the best common depth of Eq. 5 on the same data as the spectral ensemble. R^{2}NO with $M = 1$ combines all twelve iterates of the same module with the source prediction through the spectral ensemble of Section 3.3, and R^{2}NO with $M = 3$ adds two repair modules trained with different seeds and fits a single ensemble over the columns of all three.

Comparison protocol. IRNO and R^{2}NO with $M = 1$ share the frozen checkpoint, the trained repair module, and the twelve evaluations of Φ, so they differ only in how the trajectory is read out, and their comparison isolates the contribution of the spectral ensemble. The setting with $M = 3$ uses three training runs and three times this repair budget, and Section 4.2 separates its gain from the additional budget. Appendix C lists the remaining implementation choices.

Table 2: Ablation of the combination rule at a matched repair budget. All rows use the same three repair modules rolled out to $L = 12$ and differ only in how their iterates are combined. Lower is better, and the best value of each column within a backbone is set in bold.

		Cylinder		Controlled Cylinder		FSI		Foil		Combustion	
Backbone	Combination	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)
U-Net	Mean of final iterates	5.841	9.018	0.783	0.920	0.867	0.737	0.927	0.820	2.141	2.497
	Mean of all iterates	5.973	9.211	0.778	0.912	0.841	0.702	0.922	0.820	2.134	2.490
	Global combination	5.213	8.110	0.778	0.912	0.837	0.699	0.921	0.818	2.131	2.497
	R^2NO	**2.775**	**3.912**	**0.776**	**0.905**	**0.836**	**0.698**	**0.905**	**0.784**	**2.119**	**2.462**
FNO	Mean of final iterates	4.868	7.843	0.811	0.952	1.017	0.991	1.135	1.152	2.248	2.700
	Mean of all iterates	5.012	8.067	0.828	0.981	1.040	1.029	1.127	1.122	2.240	2.700
	Global combination	4.463	6.925	0.808	0.946	1.010	0.971	1.120	1.100	2.235	2.715
	R^2NO	**3.432**	**4.919**	**0.804**	**0.934**	**0.985**	**0.927**	**1.061**	**1.019**	**2.201**	**2.659**
DPOT-S	Mean of final iterates	4.080	6.501	0.793	0.937	0.866	0.729	1.051	1.147	2.100	2.331
	Mean of all iterates	4.168	6.569	0.801	0.947	0.886	0.731	1.055	1.151	2.082	2.329
	Global combination	3.829	5.990	0.793	0.931	0.864	0.729	1.048	1.139	2.071	2.333
	R^2NO	**2.617**	**3.665**	**0.789**	**0.914**	**0.860**	**0.726**	**1.016**	**1.102**	**2.058**	**2.318**

Metrics. We report the root mean squared error (RMSE) and the frequency-domain root mean squared error (fRMSE) of Takamoto et al. (2022), both on the measured channels of the test split. The fRMSE is sensitive to spectral content and therefore complements the RMSE for a method that composes its output in the Fourier domain. Appendix D reports the relative L_2 error.

Main results. R^2NO with $M = 3$ attains the lowest RMSE in all thirty pairs of Table 1, up to ties at the reported precision, and the lowest fRMSE in twenty-nine, the exception being U-Net on FSI, where the single-module setting is lower by under 0.2 percent. At a matched budget, R^2NO with $M = 1$ improves on IRNO at either depth in every pair on both metrics while reading the same trajectory. Selecting the depth of IRNO on the fitting split changes little, since the final depth is selected in fourteen of the thirty pairs and the summed RMSE and fRMSE fall by only 0.4 percent each, and summed over the thirty pairs R^2NO with $M = 1$ still lowers the RMSE of the tuned baseline by 14.5 percent and its fRMSE by 20.6 percent. The gain therefore comes from combining depths cell by cell rather than from choosing one depth well, and the larger reduction in fRMSE agrees with the diagnostic of Section 2, because a single depth over-refines high-frequency components that the ensemble can instead draw from earlier iterates or the source prediction. Adding two repair modules lowers the summed errors by a further 8.1 and 11.7 percent.

Retention of the source prediction. At its final depth, IRNO fails to improve on full finetuning in five of the thirty pairs under RMSE and in three under fRMSE, namely for U-Net on FSI, Controlled Cylinder, and Combustion and, under RMSE, for DPOT-S and FNO on Combustion, where the source prediction is already accurate and a common refinement depth degrades components that required no repair. Selecting the depth removes these failures only by halting the repair of the entire field after two to four steps, which retains less of the improvement that R^2NO with $M = 1$ obtains in the same pairs and still leaves U-Net on Combustion worse than full finetuning under fRMSE. R^2NO instead retains the source prediction only in cells where it is accurate and repairs the others, and it improves on full finetuning in every pair with either number of repair modules. This is the behavior that Proposition B.2 guarantees on the fitting split, since the all-zero weighting that reproduces the source prediction is always feasible, and it carries over to the test split in every case we evaluate.

4.2 Ablations

Combination rule at a matched budget. Table 2 asks whether the gain of R^2NO with $M = 3$ comes from the additional repair modules or from how their iterates are combined, so every row uses the same three modules and thirty-six evaluations of Φ on U-Net, FNO, and DPOT-S and differs only in the combination rule. The mean of final iterates averages the three final iterates, the mean of all iterates averages all thirty-six, and the global combination fits Eq. 14 on the single cell $N_\Omega = 1$ with the same ridge selection. Relative to full finetuning and summed over the fifteen pairs, these rules reduce the RMSE by 6.4, 5.3, and 10.9 percent and the fRMSE by 5.5, 4.4, and 11.5 percent,

whereas the full partition reaches 26.4 and 32.8 percent, and weighting every depth alike is worse than using the final depth alone. R^{2}NO attains the lowest error in every column, so the dominance that Proposition B.2 establishes on the fitting split, where both uniform averages are feasible, also holds on the test split. Removing any single axis of the partition raises the summed RMSE from 0.232 to between 0.241 and 0.248, as reported in Appendix E.

Two axes that consume no repair budget. The retained depths and the ridge parameter enter only the closed-form solve of Eq. 14, so neither adds an evaluation of Φ, and every configuration in Figures 3 and 4 consumes the same thirty-six evaluations. Figure 3 varies the number of columns J by retaining fewer depths in $\mathcal{S}$, and although the rollout passes through every intermediate iterate so that four and thirty-seven columns cost the same, the error falls throughout. Summed over the ten pairs formed by FNO and DPOT-S, the RMSE drops by 5.2 percent and the fRMSE by 9.6 percent, whereas the relative L_2 error moves by 0.4 percent, so the same axis is worth more than twenty times as much on one metric as on another. All three curves are flat beyond twenty-five columns, and we retain every depth at the operating point because nothing remains to select rather than because the last twelve columns improve accuracy.

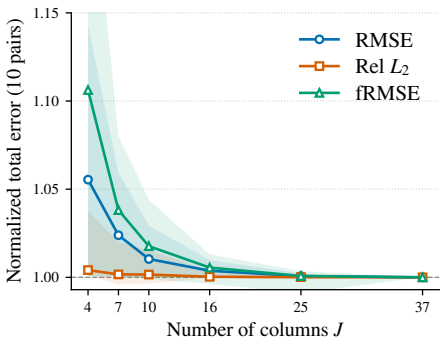


Figure 3: **Number of columns.** Each curve is normalized by its value at thirty-seven columns, and every configuration consumes thirty-six evaluations of Φ.

Figure 4 varies the ridge parameter over eight orders of magnitude. Its two extremes bracket the method, since $\lambda \to 0$ recovers the unregularized combination of Proposition B.2 and $\lambda \to \infty$ drives every weight to zero and therefore returns the source prediction, because Eq. 11 expresses the output as a correction to that prediction. The error accordingly rises by between ten and thirty-four percent across the sweep and then stops, and every curve stays below the corresponding source prediction throughout, so a poorly chosen λ degrades the combination towards its starting point rather than away from it. The value selected on a held-out half of the fitting split lies on the flat left portion of every curve and coincides with the best value on the test split, or is indistinguishable from it at the reported precision. That value is selected per system rather than per backbone, so one backbone may receive different values in different systems.

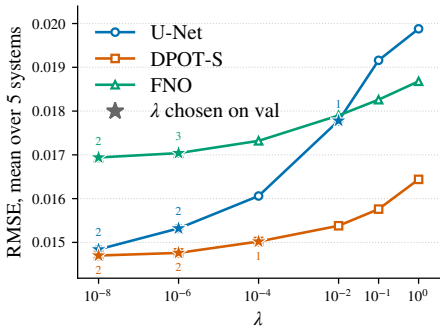


Figure 4: **Ridge parameter.** RMSE averaged over the five systems. Dotted lines mark the source prediction, and stars mark the values selected on the fitting split, with the number of systems selecting each.

5 Conclusion

We introduced R^{2}NO, which adapts a simulation-pretrained neural operator to real measurements by repairing its finetuned prediction with a shared module and composing the final field, cell by cell in the Fourier domain, from the source prediction and every iterate of the repair trajectory. On five RealPDEBench systems and six backbones, the composition consistently improves on both full finetuning and iterative refinement at the same repair budget, and the ablations attribute this gain to the cell-wise combination rather than to the additional repair modules. These results suggest that adaptation depth is better fitted per spectral component than chosen once for the entire field.

Limitations. The fixed-point analysis is local and conditional, and the dominance guarantee holds on the fitting split rather than unseen data, so the cell-wise weights depend on a fitting split of adequate size. Spectral composition minimizes error rather than enforcing physical constraints, so the composed field need not satisfy the relations the governing equations impose on its Fourier coefficients. The three-module configuration triples the training and repair computation of iterative refinement, although the single-module configuration already improves on it at the same budget.

AI USE STATEMENT

Generative AI tools were used to assist with revising the manuscript. Specifically, they were used to improve the clarity and presentation of the text during the editing process. The authors reviewed and verified all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

All experiments use the public RealPDEBench benchmark under its standard protocol. Section 3 describes the method, Appendix C.2 lists all training, fitting, and hardware settings, and Appendix C.1 describes the diagnostic of Figure 1. We will release the code and trained repair modules upon publication.

REFERENCES

Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P Kaelbling, Joshua B Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 3066–3073. IEEE, 2018.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, et al. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. *Advances in Neural Information Processing Systems*, 38:96572–96617, 2026.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pp. 2806–2823. PMLR, 2023.

Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D Collins, Michael S Pritchard, and Alexander Keller. Fourcastnet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, 2025.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Anirban Chandra, Marius Koch, Suraj Pawar, Aniruddha Panda, Kamyar Azizzadenesheli, Jeroen Snippe, Faruk O Alpak, Farah Hariri, Clement Etienam, Pandu Devarakota, et al. Fourier neural operator based surrogates for co_2 storage in realistic geologies. *arXiv preprint arXiv:2503.11031*, 2025.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 international conference on robotics and automation (ICRA)*, pp. 8973–8979. IEEE, 2019.

Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:11219–11231, 2023.

Woojin Cho, Minju Jo, Haksoo Lim, Kookjin Lee, Dongeun Lee, Sanghyun Hong, and Noseong Park. Parameterized physics-informed neural networks for parameterized pdes. *arXiv preprint arXiv:2408.09446*, 2024.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. *arXiv preprint arXiv:1910.10073*, 2019.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pp. 11398–11442. PMLR, 2023.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

Junhyuk Heo, Junghwan Park, Sangcheol Sim, Beomkyu Choi, and Woojin Cho. Flame: Physics-guided neural operators for onboard satellite methane detection in hyperspectral imagery. *arXiv preprint arXiv:2606.01577*, 2026.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Jun-Ting Hsieh, Shengjia Zhao, Stephan Eismann, Lucia Mirabella, and Stefano Ermon. Learning neural pde solvers with convergence guarantees. *arXiv preprint arXiv:1906.01200*, 2019.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, et al. Realpdebench: A benchmark for complex physical systems with real-world data. *arXiv preprint arXiv:2601.01829*, 2026.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. Loopformer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. *arXiv preprint arXiv:2602.11451*, 2026.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Xiaotian Liu, Shuyuan Shang, Xiaopeng Wang, Pu Ren, and Yaoqing Yang. Iterative refinement neural operators are learned fixed-point solvers: A principled approach to spectral bias mitigation. *arXiv preprint arXiv:2605.24041*, 2026.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Tanya Marwah, Ashwini Pokle, J Zico Kolter, Zachary Lipton, Jianfeng Lu, and Andrej Risteski. Deep equilibrium based neural operators for steady-state pdes. *Advances in Neural Information Processing Systems*, 36:15716–15737, 2023.

Sajad Movahedi, Vera Milovanović, Shlomo Libo Feigin, Alexander Theus, Thomas Hofmann, Valentina Boeva, T Konstantin Rusch, and Antonio Orvieto. Fixed-point reasoners: Stable and adaptive deep looped transformers. *arXiv preprint arXiv:2606.18206*, 2026.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810. IEEE, 2018.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel De Bézenac. Convolutional neural operators for robust and accurate learning of pdes. *Advances in Neural Information Processing Systems*, 36: 77187–77200, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, volume 2025, pp. 14855–14881, 2025.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in neural information processing systems*, 35:1596–1611, 2022.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Gege Wen, Zongyi Li, Qirui Long, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. Real-time high-resolution co2 geological storage prediction using nested fourier neural operators. *Energy & Environmental Science*, 16(4):1732–1741, 2023.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Huaiqian You, Quinn Zhang, Colton J Ross, Chung-Hao Lee, and Yue Yu. Learning deep implicit fourier neural operators (ifnos) with applications to heterogeneous material modeling. *Computer Methods in Applied Mechanics and Engineering*, 398:115296, 2022.

Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4): 1307–1319, 2020.

A Related Work

Neural operators from simulation to measurements. Neural operators such as FNO (Li et al., 2020), DeepONet (Lu et al., 2021), CNO (Raonic et al., 2023), and Transolver (Wu et al., 2024) learn solution maps of PDEs from simulated fields and now serve as surrogates in weather forecasting (Pathak et al., 2022; Bonev et al., 2025; 2023), satellite methane detection (Heo et al., 2026), subsurface carbon storage (Wen et al., 2023; Chandra et al., 2025), and materials mechanics (You et al., 2022). Pretraining on many simulated systems, as in DPOT (Hao et al., 2024), yields representations that transfer across systems, but still within simulation. RealPDEBench (Hu et al., 2026) pairs such simulations with real measurements of the same systems, and its real-world finetuning task is the setting we adopt. Finetuning on a few real trajectories is the natural first step, but it can overwrite structure that pretraining had captured. A two-phase remedy, pretraining across a family of PDEs and then adapting only a handful of coefficients to a target instance, has been explored for physics-informed neural networks (Cho et al., 2023; 2024). R^{2}NO follows the two-phase algorithm in the output space. It finetunes first, then corrects the prediction of the finetuned operator with a repair module and decides, cell by cell in the Fourier domain, how much of that prediction to keep.

Iterative refinement and the combination of its iterates. Reusing one block over many steps separates computational depth from parameter count. Looped transformers unroll a weight-shared block for a fixed or budgeted number of iterations (Giannou et al., 2023; Saunshi et al., 2025), LoopFormer trains on trajectories of varying length so that the budget can be chosen at test time (Jeddi et al., 2026), and deep equilibrium models run the iteration to its fixed point (Bai et al., 2019; Marwah et al., 2023). IRNO brings this structure to neural operators as deterministic residual corrections of a frozen prediction, analyzed as a fixed-point iteration (Liu et al., 2026). Adaptive-computation methods choose the number of steps per input or per token (Graves, 2016), but every one of these methods reads out a single depth for the unit it adapts, whereas R^{2}NO keeps every iterate as a candidate. Combining candidates with weights fitted on held-out data is stacking (Wolpert, 1992; Breiman, 1996), and averaging checkpoints along a training trajectory is a snapshot ensemble (Huang et al., 2017). R^{2}NO stacks the finetuned prediction and the iterates of several repair modules with one set of weights per Fourier cell rather than one for the whole field.

The sim-to-real gap. Closing the gap between simulation and the real world matters across engineering, because simulators idealize the systems they model while measurements carry noise, unmodeled effects, and conditions the simulation never saw. Robotics has addressed this gap from the simulation side, by randomizing simulator parameters so that a policy becomes robust to the mismatch (Tobin et al., 2017; Peng et al., 2018) or by calibrating the simulator against real rollouts (Chebotar et al., 2019), and from the model side, by learning a residual that corrects the simulator's predictions on real data (Ajay et al., 2018; Zeng et al., 2020). The latter idea is older in statistics, where a calibrated computer model is completed by an additive discrepancy term fitted to observations (Kennedy & O'Hagan, 2001). R^{2}NO pursues the same goal for scientific machine learning, where neural surrogates of physical systems are trained on simulation and deployed on measurements. Its repair module is a discrepancy model applied to the prediction of a finetuned operator, and its spectral ensemble decides where the discrepancy needs correcting and where the source prediction already transfers.

B Analysis of Repair and Component Selection

This appendix collects the analysis behind Sections 2 and 3: convergence of the repair loop versus accuracy of its iterates, dominance of the fitted ensemble on the fitting split, and generalization of a cell-wise selection. Throughout, $\|\cdot\|_2$ is the Euclidean norm of a vectorized field, P_b are the projections of Eq. 2, and the member index of Eq. 6 is dropped.

B.1 Convergence of the repair loop and accuracy of its iterates

Fix an input history $\mathcal{X}$ with target y and write the loop of Eq. 6 as $h_{\ell+1} = T(h_\ell)$ with $T(h) := h + \alpha\,\Phi(\mathcal{X}, h; \theta)$. A fixed point $h^\star = T(h^\star)$ is a prediction at which the learned correction vanishes. Let $d := \|T(y) - y\|_2 = \alpha\|\Phi(\mathcal{X}, y; \theta)\|_2$ be the correction the module would still apply at the target, the quantity that the fixed-point penalty of Eq. 8 penalizes, and assume that T is a contraction with

factor $q < 1$ on a ball $\overline{B}_r(y)$ of radius r around the target that contains h_0, with $d \le (1-q)r$. Neither pretraining nor the fixed-point penalty proves these assumptions; the viewpoint is classical and is also used for learned refinement by Bai et al. (2019); Marwah et al. (2023); Liu et al. (2026).

Proposition B.1 (Local convergence and target bias)**.** *Under these assumptions the iterates converge to the unique fixed point $h^\star$ of T in $\overline{B}_r(y)$, and with $e_\ell := \|h_\ell - y\|_2$,*

$$e_\ell \le q^\ell e_0 + \frac{1-q^\ell}{1-q}\, d \quad \textit{for every } \ell \ge 0, \qquad \frac{d}{1+q} \le \|h^\star - y\|_2 \le \frac{d}{1-q}. \tag{15}$$

Proof. For $h \in \overline{B}_r(y)$, $\|T(h) - y\|_2 \le q\|h - y\|_2 + d \le r$, so T maps the ball into itself and Banach's theorem gives the fixed point. The same inequality gives $e_{\ell+1} \le q e_\ell + d$, which iterates to the first bound and, as $\ell \to \infty$, to the upper bias bound. The lower bound follows from $d \le \|T(y) - T(h^\star)\|_2 + \|h^\star - y\|_2 \le (1+q)\|h^\star - y\|_2$. □

The first term contracts the error of the source prediction and the second is the price of an imperfect correction at the target. When $d > 0$, an extra step is guaranteed to help only while $e_\ell > d/(1-q)$, and if $h_0 = y$ the first repair moves away from the target although the loop is contractive, which is why the source prediction enters the ensemble as its own column. Convergence also says nothing about where each region attains its best error. Take $y = (0,0)$, $h_0 = (-1,-3)$, and $T(h) = \frac{1}{2}h + \frac{1}{2}(1,1)$, a $\frac{1}{2}$-contraction with $h^\star = (1,1)$, $h_1 = (0,-1)$, and $h_2 = (\frac{1}{2}, 0)$. With P_1 and P_2 the projections onto the two coordinates,

$$\|P_1 h_1 + P_2 h_2 - y\|_2^2 = 0 < \tfrac{1}{4} = \min_{\ell \ge 0} \|h_\ell - y\|_2^2, \tag{16}$$

so the region-wise composition is exact while no single iterate is, and the composed field is not a fixed point of T. This is what Eq. 5 measures on real data.

B.2 Dominance of the fitted combination on the fitting split

For any weight array $\tilde{w} = (\tilde{w}_{b,j})$, let $\widehat{Q}(\tilde{w})$ be the fitting error of Eq. 12 with $\tilde{w}$ in place of the fitted weights, and $\widehat{Q}_b(\tilde{w}_b)$ its term for cell b, so that $\widehat{Q}(\tilde{w}) = \sum_b \widehat{Q}_b(\tilde{w}_b)$.

Proposition B.2 (Dominance on the fitting split)**.** *Let $w^\star$ minimize $\widehat{Q}$; the weights of Eq. 14 with $\lambda = 0$ do so whenever every $\mathbf{G}_b$ is invertible. Then $\widehat{Q}(w^\star) \le \widehat{Q}(\tilde{w})$ for every $\tilde{w}$, in particular for the source prediction ($\tilde{w} = 0$), any single candidate ($\tilde{w}_b = e_j$ for all b, with e_j the one-hot vector selecting column j), and any cell-wise selection ($\tilde{w}_b = e_{j(b)}$).*

The proposition holds because each $\widehat{Q}_b$ depends only on $\tilde{w}_b$, so $\widehat{Q}$ is minimized by minimizing every cell separately, which is what the normal equations of Eq. 13 do, while any other $\tilde{w}$ is merely one feasible choice in every cell. For $M = 1$ and $\mathcal{S} = \{1, \dots, L\}$, a cell-wise selection is the composite field $\sum_b P_b h_{\ell_b}$ of Section 2, so the common-depth gap of Eq. 5 is the one-hot special case of the proposition. Since the source prediction is that of the full finetuning baseline, the case $\tilde{w} = 0$ shows that the ensemble cannot fit worse than that baseline. The argument covers the squared error of Eq. 12, but not the relative error or frequency-bin aggregates of the benchmark, and the ridge term gives up part of the inequality for stability.

B.3 Generalization of a cell-wise selection

Proposition B.2 concerns the fitting split. For $M = 1$ and $\mathcal{S} = \{1, \dots, L\}$, let the fitting split consist of n real trajectories, treated as the independent units because windows within a trajectory overlap, and let $\widehat{E}_b^{\mathrm{fit}}(\ell)$ be the regional error of Eq. 3 on the fitting split, an average over trajectories of a per-trajectory error in $[0, C_b]$ whose expectation over the real distribution is $E_b(\ell)$. The repair module is trained on $\mathcal{D}^{\mathrm{train}}$, and one of N_{ckpt} saved checkpoints is selected on the fitting split ($N_{\mathrm{ckpt}} = 12$ in our experiments), so the candidates of every checkpoint are independent of the fitting split. Hoeffding's inequality and a union bound over the $N_\Omega(L+1)N_{\mathrm{ckpt}}$ combinations of region, depth, and checkpoint give, with probability at least $1 - \delta$ and simultaneously for every saved checkpoint, hence for the one selected on the fitting split,

$$\left|E_b(\ell) - \widehat{E}_b^{\mathrm{fit}}(\ell)\right| \le \xi_b := C_b \sqrt{\frac{\log\!\left(2N_\Omega(L+1)N_{\mathrm{ckpt}}/\delta\right)}{2n}} \qquad \text{for all } b \text{ and } \ell \in \mathcal{S}_0. \tag{17}$$

On this event a minimizer ℓ_b of $\widehat{E}_b^{\mathrm{fit}}$ satisfies $E_b(\ell_b) \le \min_\ell E_b(\ell) + 2\xi_b$, and since Eq. 4 makes the risk additive across regions, the composite field $\sum_b P_b h_{\ell_b}$ has population risk at most $\sum_b \min_{\ell \in \mathcal{S}_0} E_b(\ell) + 2\sum_b \xi_b$. Region-wise selection thus pays $2\xi_b$ per region for choosing on data, a price that grows only logarithmically in the number of regions, depths, and checkpoints. Any other choice made on the fitting split, such as the ridge parameter, must be covered by the union bound or made on a separate split, and the fitted weights of Eq. 14, which range over a continuous class, would need a uniform bound that we do not pursue.

C Experimental Setup

This appendix gives the details of the experiments in Section 4, including the setup of the diagnostic in Figure 1 and the data, models, training, fitting, and hardware settings behind Tables 1 and 2.

C.1 Frequency-resolved diagnostic of Figure 1

Model and data. The diagnostic uses the Cylinder system of RealPDEBench, with the finetuned FNO checkpoint of Section 4 as the frozen source operator $\Psi(\cdot;\pi)$. The refinement module $\Phi(\cdot;\theta)$ is trained on $\mathcal{D}^{\mathrm{train}}$ with the objective of Eq. 8 and the architecture, step size, and optimization settings of the IRNO baseline. After training, π and θ are held fixed and Eq. 1 is run for $L = 12$ steps on every input history of the test split $\mathcal{D}^{\mathrm{test}}$, which yields the iterates $h_0, \dots, h_L$. No hyperparameter or design choice of Section 3 is selected on this split.

Metric. Let $|\Omega_b|$ denote the number of coefficients retained by the mask of region b across measured channels and forecast frames. The fRMSE of region b at depth ℓ and its percentage change relative to depth zero, which Figure 1(c) reports, are

$$\mathrm{fRMSE}_b(\ell) = \left(\widehat{\mathbb{E}}_{\mathrm{test}} \tfrac{1}{|\Omega_b|} \left\| \mathbf{1}_{\Omega_b} \odot \mathcal{F}(h_\ell - y) \right\|_2^2 \right)^{1/2}, \tag{18}$$

$$\Delta\,\mathrm{fRMSE}_b(\ell) = 100\,\frac{\mathrm{fRMSE}_b(\ell) - \mathrm{fRMSE}_b(0)}{\mathrm{fRMSE}_b(0)}. \tag{19}$$

Negative values indicate improvement. Because $\mathcal{F}$ is unitary, the regional squared error of Eq. 3 satisfies $\widehat{E}_b(\ell) = |\Omega_b|\,\mathrm{fRMSE}_b(\ell)^2$, so the markers in Figure 1(c), which mark the minimum over $\ell \in \mathcal{S}_0$, are also the region-wise minimizers in Eq. 5.

C.2 Training and fitting details

Data and splits. We use the real-world finetuning task of RealPDEBench for all five systems. The training split of the benchmark serves as $\mathcal{D}^{\mathrm{train}}$ for the repair modules, its validation split serves in full as the fitting split $\mathcal{D}^{\mathrm{fit}}$ of the spectral ensemble, and its test split is used only to compute the reported errors. Each output window contains 20 forecast frames, except for Controlled Cylinder with 10, on a grid of 64×128 points for Cylinder, Controlled Cylinder, and Foil and 64×64 points for FSI and Combustion. The measured channels are read from the benchmark data and are two of the three reported channels for the fluid systems and one of the sixteen reported channels for Combustion.

Source operators. For U-Net, DeepONet, FNO, Transolver, and DPOT-S, the source operator is the finetuned checkpoint released with RealPDEBench. For CNO we finetune the simulation-pretrained backbone ourselves under the benchmark protocol, because the released checkpoint does not work. The source operator is frozen after finetuning, and its prediction h_0 is computed once per window and shared by every repair module.

Repair modules. The repair module Φ is a U-Net with base width 32 that receives the input history and the current iterate, concatenated over all forecast frames along the channel axis, and returns an update with the shape of the output field, which amounts to about 8.7 million parameters for Cylinder. It is trained for 12 epochs with learning rate 3×10^{-4} and step size $\alpha = 0.2$, supervising 12 iterates of the loop with the objective of Eq. 8. The spatial term is the mean squared error over the measured channels, the spectral term compares the magnitudes of the real-input Fourier coefficients normalized by the number of grid points with the exponent η_ℓ increasing linearly from one to two, and the

fixed-point term is evaluated once per batch at the target, with weights $\beta_{\mathrm{spe}} = 1.0$ and $\beta_{\mathrm{fp}} = 0.01$. The checkpoint is selected among the twelve saved epochs by the error on the validation split, which is also the fitting split of the ensemble, and Appendix B.3 accounts for this dependence.

Iterative refinement baseline. IRNO uses the repair module trained with a fixed seed of 42 and is reported at depth $L = 12$ and, in Table 1, at the depth $\ell^\star$ selected on the fitting split, so it shares the trained trajectory with R^{2}NO with $M = 1$ and differs from it only in the readout. The loss terms, their weights, the step size, and the learning rate follow the published configuration of the baseline. Its number of epochs, base width, and padding differ from that configuration because the benchmark differs, and these differences, together with the normalization of the Fourier transform in the spectral term and the absence of a warmup on the fixed-point term, apply identically to IRNO and to R^{2}NO.

Spectral ensemble. The Fourier transform is applied over the two spatial axes of the denormalized output field, and the cells of Eq. 9 use $N_\rho = 128$ radial bands and $N_\varphi = 16$ angular sectors, shared across forecast frames. Because the partition is finer than the Fourier grid near the origin, $1{,}468$ of the $2{,}048$ requested spatial cells are occupied on the 64×128 grid and $1{,}182$ on the 64×64 grid, and empty cells receive zero weight. With all twelve depths retained, the ensemble has $J = 13$ columns for $M = 1$ and $J = 37$ for $M = 3$, which gives $151{,}552$ weights for Cylinder with $M = 3$. The Gram matrices and right-hand sides of Eq. 13 are accumulated in double precision over every window of the fitting split, each window entering with weight $\|y\|_2^{-1}$, a fixed reweighting of Eq. 12 that lies between the unweighted squared error and the relative error and leaves Proposition B.2 unchanged. The ridge parameter is selected from $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1\}$ by solving on one half of the fitting split and scoring on the other half, after which the system is solved again on the full split. Each cell is fitted from many more observations than it has weights, since for Cylinder the fitting split contains $4{,}820$ windows and even a cell with a single Fourier coefficient accumulates $96{,}400$ observations for its 37 weights.

Computational cost. Training a repair module and fitting the ensemble are the only adaptation steps, and the latter is a closed-form solve that requires no gradient updates. At inference, R^{2}NO with $M = 1$ evaluates Φ twelve times, exactly as IRNO does, and R^{2}NO with $M = 3$ evaluates it thirty-six times. The fitted weights are expanded once into a tensor over Fourier coefficients, so the ensemble adds only one forward and one inverse transform per window to the cost of the rollout. Experiments run on a server with four NVIDIA RTX A6000 GPUs of 48 GB each, two AMD EPYC 9224 processors with 48 cores in total, and 377 GB of memory, using PyTorch 2.7.0 with CUDA 12.6 in single precision, while the Gram matrices of the spectral ensemble are accumulated in double precision. We use version 0.1.0 of the RealPDEBench code at commit `62f4c80` and version 2.0.0 of its data.

D Additional results

Table 3 reports the relative L_2 error for every setting of Table 1. R^{2}NO with $M = 3$ improves on full finetuning in twenty-nine of the thirty pairs and R^{2}NO with $M = 1$ in twenty-eight, compared with twenty-two for IRNO. At the matched budget of twelve evaluations of Φ, R^{2}NO with $M = 1$ improves on IRNO in twenty-six pairs, and the four exceptions all lie on Cylinder, where IRNO attains the lowest relative L_2 error for U-Net, CNO, and DPOT-S and full finetuning does so for FNO. This metric normalizes each window by the norm of its target and therefore weights windows differently from the squared error that the combination minimizes on the fitting split. On Cylinder a small fraction of windows carries most of the squared error, so a combination fitted to that error need not be optimal under the relative normalization, while on the remaining four systems R^{2}NO with $M = 3$ attains the lowest relative L_2 error for every backbone.

E Additional ablations

Axes of the partition. Table 4 removes one axis of the partition at a time while keeping the fitted combination, the three repair modules, and the repair budget of the operating point. Summed over the fifteen pairs, the RMSE rises from 0.2323 for the full partition to 0.2408 without the angular axis, 0.2478 without the radial axis, and 0.2446 without the channel axis, against 0.2812 for a single cell,

Table 3: Relative L_2 error ($\times 10^{-2}$, lower is better) on RealPDEBench for the settings of Table 1. The repair budget counts evaluations of Φ, and the best value of each column within a backbone is set in bold.

Backbone	Adaptation	Repair budget	Cylinder	Controlled Cylinder	FSI	Foil	Combustion
U-Net	Full finetuning	0	7.284	5.435	5.789	1.447	54.025
	IRNO	12	**6.926**	5.421	6.053	1.454	55.257
	R^{2}NO (M=1)	12	7.652	5.317	5.753	1.414	53.840
	R^{2}NO (M=3)	36	7.103	**5.314**	**5.751**	**1.412**	**53.789**
DeepONet	Full finetuning	0	15.031	22.837	23.675	3.751	57.215
	IRNO	12	8.468	6.678	7.640	2.869	55.707
	R^{2}NO (M=1)	12	8.395	6.041	6.772	2.436	55.147
	R^{2}NO (M=3)	36	**7.745**	**5.848**	**6.429**	**2.122**	**54.267**
CNO	Full finetuning	0	8.298	5.672	6.788	2.055	58.435
	IRNO	12	**7.060**	5.517	6.054	2.115	56.517
	R^{2}NO (M=1)	12	7.384	5.490	6.032	1.719	55.716
	R^{2}NO (M=3)	36	7.063	**5.482**	**5.972**	**1.660**	**54.963**
FNO	Full finetuning	0	**7.804**	7.025	8.807	2.056	56.793
	IRNO	12	7.940	5.784	6.927	2.159	57.258
	R^{2}NO (M=1)	12	7.994	5.745	6.786	1.767	55.579
	R^{2}NO (M=3)	36	7.839	**5.673**	**6.584**	**1.734**	**55.387**
Transolver	Full finetuning	0	17.063	11.653	14.900	3.352	78.024
	IRNO	12	8.791	6.221	7.120	2.101	64.365
	R^{2}NO (M=1)	12	7.992	5.959	6.758	1.867	56.295
	R^{2}NO (M=3)	36	**7.403**	**5.769**	**6.650**	**1.839**	**55.381**
DPOT-S	Full finetuning	0	7.660	6.151	7.007	1.733	53.786
	IRNO	12	**6.919**	5.599	6.043	1.684	55.378
	R^{2}NO (M=1)	12	7.379	5.533	6.019	1.634	53.152
	R^{2}NO (M=3)	36	6.941	**5.470**	**5.984**	**1.623**	**53.028**

so each axis recovers part of the gap and none of them accounts for it alone. The fRMSE shows the same ordering more sharply, rising from 0.02794 to 0.02917, 0.03150, and 0.03105 respectively, against 0.03682 for a single cell, so the radial axis matters most for the frequency-domain error, consistent with the dependence of the best depth on frequency magnitude in Section 2. The effect of the individual axes is concentrated on Cylinder and lies at or below the reported precision on the remaining systems. Removing the channel axis leaves Combustion unchanged, because that system has a single measured channel and the corresponding rows are identical by construction.

Base column. Table 5 removes the base column $-h_0$ of Eq. 10. Without it, the prediction in each cell is an affine combination of the source prediction and the retained iterates, and it still reproduces the source prediction when every weight vanishes, because Eq. 11 expresses the output as a correction to that prediction. The base column lifts the constraint that the coefficients sum to one and thereby lets a cell shrink its overall amplitude, which reduces the squared error in cells where every candidate carries more energy than the target. Removing it raises the summed RMSE over the fifteen pairs from 0.2323 to 0.2378, and at full precision the RMSE increases in every pair, with the largest change of 10.9 percent for DPOT-S on Cylinder. The summed fRMSE rises from 0.02794 to 0.02818, although the column slightly increases the fRMSE in five pairs, most visibly by 1.8 percent for DPOT-S on Foil, since it is fitted to the squared error on the fitting split and need not reduce the frequency-domain error in every pair. The combination without the column still improves on full finetuning in every pair, which confirms that retention of the source prediction follows from the residual form of the output rather than from the column.

Table 4: Removing one axis of the partition at a matched repair budget. All rows use three repair modules rolled out to $L = 12$ and fitted weights, and differ only in the partition. The RMSE and the fRMSE are reported, lower being better, and the best value of each column within a backbone is set in bold, with ties at the reported precision marked jointly.

		Cylinder		Controlled Cylinder		FSI		Foil		Combustion	
Backbone	Partition	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)
U-Net	Single cell	5.213	8.110	0.778	0.912	0.837	0.699	0.921	0.818	2.131	2.497
	No radial axis	3.606	5.709	**0.775**	0.905	**0.836**	**0.697**	0.908	0.794	2.121	2.468
	No angular axis	3.154	4.473	0.776	**0.904**	**0.836**	**0.697**	0.910	0.793	2.122	2.468
	No channel axis	3.412	5.557	0.776	0.908	**0.836**	**0.697**	0.910	0.798	**2.119**	**2.462**
	Full partition	**2.775**	**3.912**	0.776	0.905	**0.836**	0.698	**0.905**	**0.784**	**2.119**	**2.462**
FNO	Single cell	4.463	6.925	0.808	0.946	1.010	0.971	1.120	1.100	2.235	2.715
	No radial axis	3.810	5.743	0.805	0.937	0.986	0.928	1.072	1.040	2.204	2.665
	No angular axis	3.682	5.260	0.805	**0.934**	0.989	0.929	1.081	1.045	2.206	2.663
	No channel axis	3.707	5.537	0.805	0.937	0.991	0.939	1.076	1.048	**2.201**	**2.659**
	Full partition	**3.432**	**4.919**	**0.804**	**0.934**	**0.985**	**0.927**	**1.061**	**1.019**	**2.201**	**2.659**
DPOT-S	Single cell	3.829	5.990	0.793	0.931	0.864	0.729	1.048	1.139	2.071	2.333
	No radial axis	2.931	4.536	**0.789**	0.917	**0.860**	**0.726**	1.022	1.111	2.059	2.320
	No angular axis	2.792	3.933	**0.789**	**0.912**	0.861	**0.726**	1.020	1.110	2.060	2.319
	No channel axis	2.891	4.413	**0.789**	0.913	0.861	**0.726**	1.028	1.133	**2.058**	**2.318**
	Full partition	**2.617**	**3.665**	**0.789**	0.914	**0.860**	**0.726**	**1.016**	**1.102**	**2.058**	**2.318**

Table 5: Effect of the base column at the operating point. Both rows use three repair modules rolled out to $L = 12$, fitted weights, and the full partition. The RMSE and the fRMSE are reported, lower being better, and the better value of each pair is set in bold, with ties at the reported precision marked jointly.

		Cylinder		Controlled Cylinder		FSI		Foil		Combustion	
Backbone	Base column	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)	RMSE ($\times 10^{-2}$)	fRMSE ($\times 10^{-3}$)
U-Net	Without	2.805	3.953	**0.776**	**0.904**	**0.836**	**0.697**	0.911	0.788	2.128	2.480
	With	**2.775**	**3.912**	**0.776**	0.905	**0.836**	0.698	**0.905**	**0.784**	**2.119**	**2.462**
FNO	Without	3.534	4.984	0.806	**0.934**	0.988	**0.927**	1.091	1.021	2.222	2.678
	With	**3.432**	**4.919**	**0.804**	**0.934**	**0.985**	**0.927**	**1.061**	**1.019**	**2.201**	**2.659**
DPOT-S	Without	2.938	3.765	0.790	**0.914**	0.862	**0.726**	1.024	**1.083**	2.068	2.323
	With	**2.617**	**3.665**	**0.789**	**0.914**	**0.860**	**0.726**	**1.016**	1.102	**2.058**	**2.318**

F ADAPTATION FROM THE SIMULATION-PRETRAINED CHECKPOINT

We also evaluate the setting in which the backbone is never finetuned: the simulation-pretrained operator is frozen and provides the source prediction h_0, and IRNO and R^{2}NO are trained and fitted on top of it with the settings of Appendix C.2. Table 6 reports the results for FNO and DPOT-S, with full finetuning shown for reference only; its prediction is not a candidate of the ensemble.

Table 6: Adaptation from the frozen simulation-pretrained checkpoint. A dagger marks a value lower than full finetuning of the same backbone. Lower is better, and the best value of each column within a backbone is set in bold.

		Cylinder		Controlled Cylinder		FSI		Foil		Combustion	
Backbone	Adaptation	RMSE ($\times10^{-2}$)	fRMSE ($\times10^{-3}$)	RMSE ($\times10^{-2}$)	fRMSE ($\times10^{-3}$)	RMSE ($\times10^{-2}$)	fRMSE ($\times10^{-3}$)	RMSE ($\times10^{-2}$)	fRMSE ($\times10^{-3}$)	RMSE ($\times10^{-2}$)	fRMSE ($\times10^{-3}$)
FNO	Pretrained (source)	7.389	11.481	2.845	5.934	4.263	5.899	2.738	3.733	3.630	5.147
	Full finetuning	5.449	8.661	0.943	1.077	1.266	1.162	1.199	1.241	**2.246**	**2.726**
	IRNO (ℓ=12)	3.194†	4.575†	0.890†	1.124	0.981†	0.927†	1.546	1.794	2.654	3.482
	R^{2}NO (M=1)	2.787†	3.545†	0.873†	1.053†	0.935†	0.818†	1.177†	1.309	2.521	3.301
	R^{2}NO (M=3)	**2.456**†	**2.981**†	**0.843**†	**0.997**†	**0.912**†	**0.773**†	**1.114**†	**1.151**†	2.347	2.932
DPOT-S	Pretrained (source)	5.124	7.595	2.333	3.939	2.594	3.054	2.210	2.712	3.932	5.418
	Full finetuning	4.404	6.726	0.847	0.987	0.990	**0.737**	1.093	1.178	**2.106**	**2.408**
	IRNO (ℓ=12)	3.266†	4.260†	0.909	1.229	1.087	1.063	1.361	1.571	2.664	3.530
	R^{2}NO (M=1)	2.796†	3.430†	0.839†	1.039	1.013	0.913	1.079†	1.125†	2.490	3.250
	R^{2}NO (M=3)	**2.258**†	**2.544**†	**0.814**†	**0.964**†	**0.951**†	0.806	**1.022**†	**0.986**†	2.446	3.161

All adapted methods improve on the source prediction in every pair, as Proposition B.2 guarantees for R^{2}NO on the fitting split. At the same repair budget, R^{2}NO with $M = 1$ improves on IRNO in every pair, lowering the summed RMSE by 11.0 percent and the summed fRMSE by 16.0 percent. It also surpasses full finetuning in seven pairs under RMSE, against four for IRNO, and R^{2}NO with $M = 3$ does so in eight, without updating any parameter of the pretrained operator. The main exception is Combustion, where full finetuning remains best.

G QUALITATIVE RESULTS

Figures 5–9 compare the predictions of DPOT-S on one test window of each system at five forecast frames. The rows show the measured target, full finetuning, IRNO, and R^{2}NO with $M = 3$, and all rows of a figure share one colour scale. At this scale the adapted methods differ only slightly, in line with the errors of Table 1, and the difference is most visible in the background of Foil, where full finetuning and IRNO carry a grid-aligned high-frequency pattern that R^{2}NO largely removes.

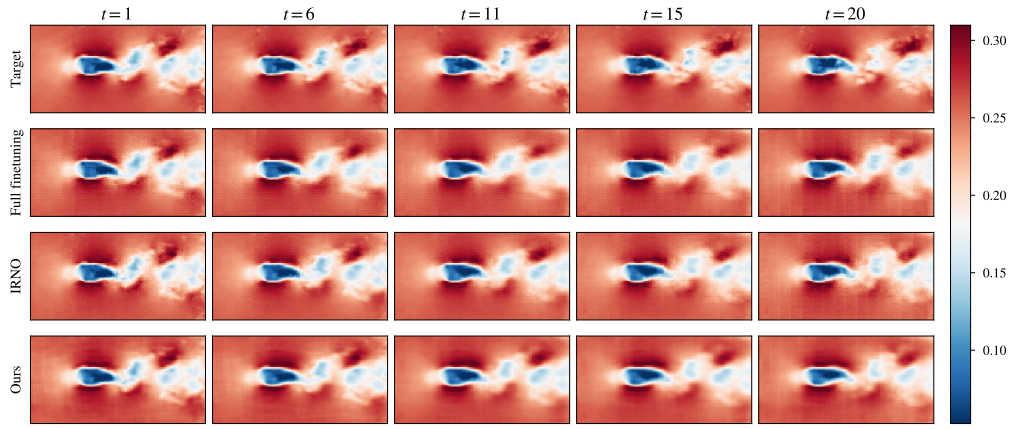


Figure 5: Predictions of DPOT-S on one Cylinder test window at five forecast frames. Rows show the measured target, full finetuning, IRNO, and R^{2}NO; all panels share one color scale.

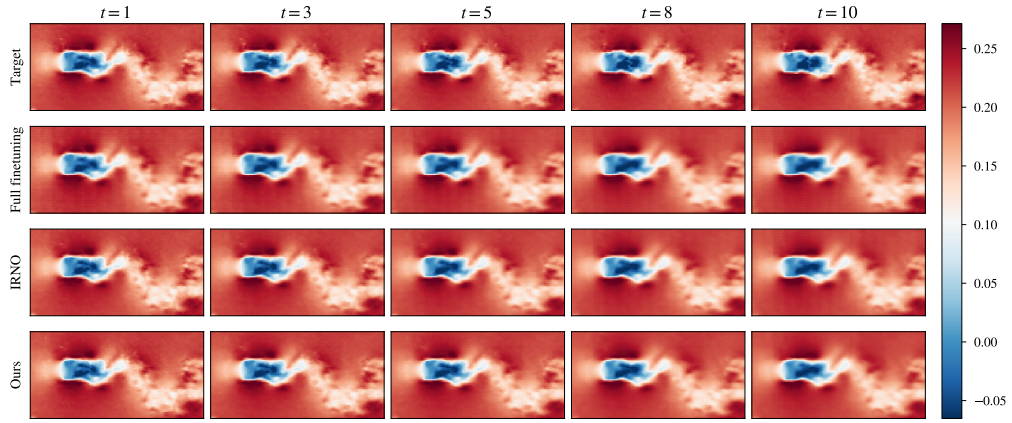


Figure 6: Predictions of DPOT-S on one Controlled Cylinder test window at five forecast frames. Rows show the measured target, full finetuning, IRNO, and R^{2}NO; all panels share one color scale.

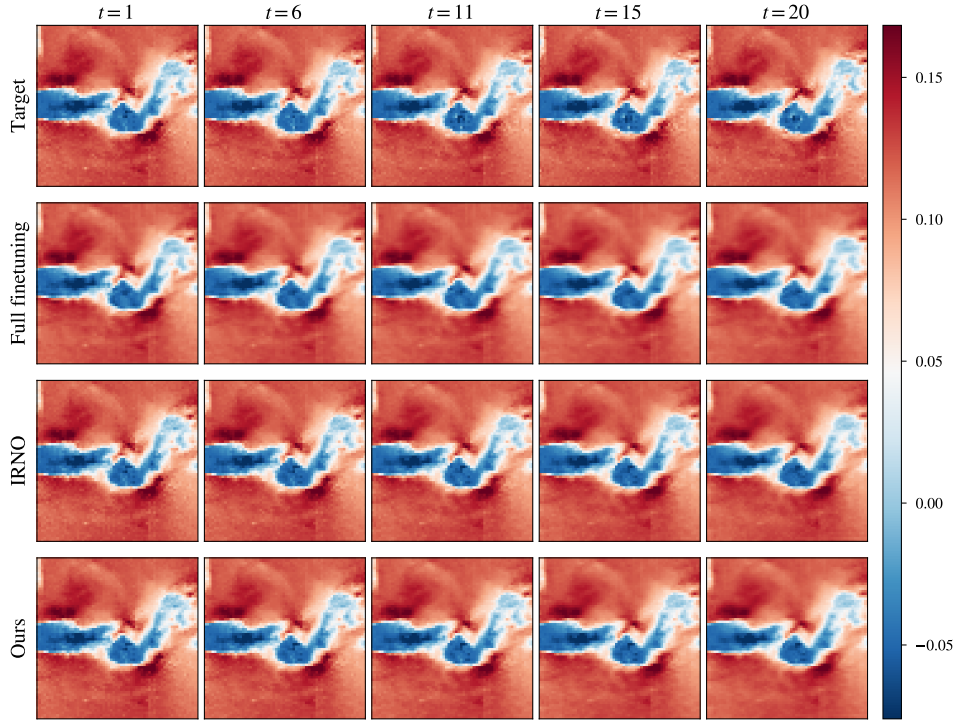


Figure 7: Predictions of DPOT-S on one FSI test window at five forecast frames. Rows show the measured target, full finetuning, IRNO, and R^{2}NO; all panels share one color scale.

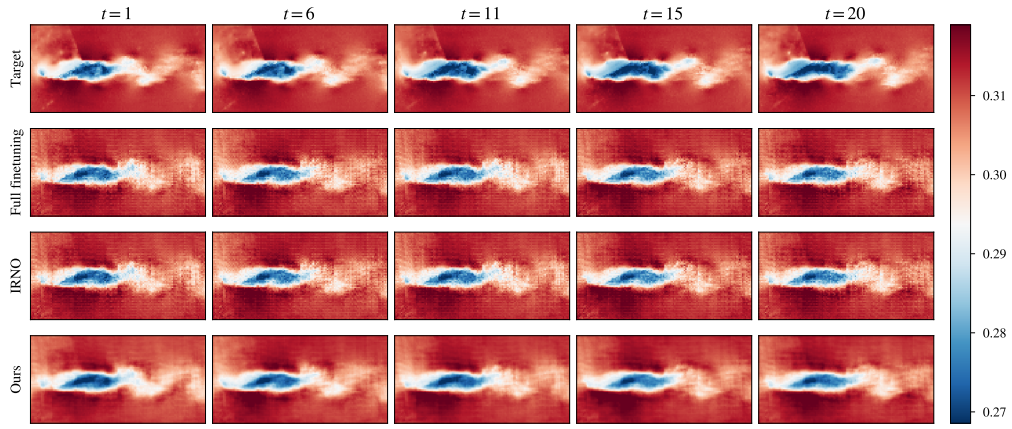


Figure 8: Predictions of DPOT-S on one Foil test window at five forecast frames. Rows show the measured target, full finetuning, IRNO, and R^2NO; all panels share one color scale.

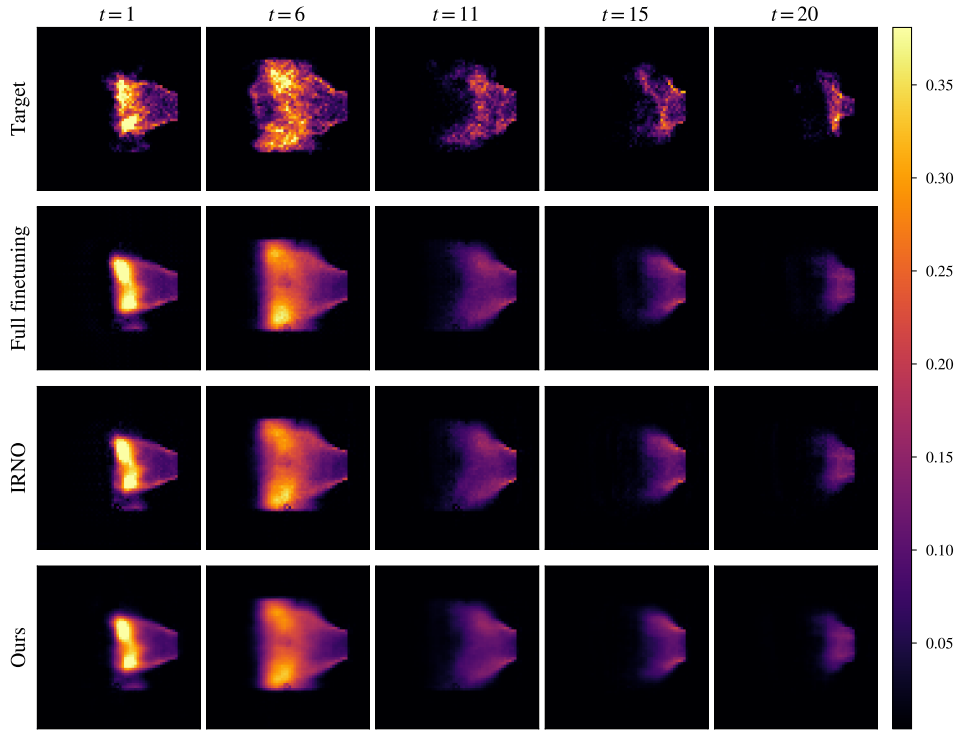


Figure 9: Predictions of DPOT-S on one Combustion test window at five forecast frames. Rows show the measured target, full finetuning, IRNO, and R^2NO; all panels share one color scale.