# AUTOMF: AUTOMATED ENSEMBLES FOR MULTIFIDELITY FIELD PREDICTION

**Anonymous authors**
Paper under anonymous review

## ABSTRACT

Learning to predict physical fields requires fine simulation data that can be expensive to obtain. Multifidelity learning combines coarse and fine solutions, but choosing a surrogate from limited fine data can discard useful predictions. We introduce AUTOMF, an automated ensemble workflow built around nine surrogates adapted from neural operators, convolutional networks, and reduced basis models. The library includes transfer learning, pooled fidelity training, and coarse field correction, together with a fine only reference. We assemble 22 datasets spanning seven classes of partial differential equations (PDEs) and climate emulation to evaluate whether combining predictions reduces errors in model choice. Five additional input and fine solution pairs, reserved from training, determine one weight per model using established ensemble rules. Weights stay fixed across the field and new inputs, which require no additional coarse simulation. The fitted mixture achieves lower error on 17 of 22 datasets than selecting one model from the same fitting examples. It reduces geometric mean relative $L^2$ error by 11.6% under equal weighting of the seven PDE classes and by 7.5% under equal weighting of all 22 datasets. Against the best implemented baseline chosen separately for each PDE dataset after evaluation, its reduction is 29.4% under equal weighting of the seven PDE classes. Code and reproducibility materials are available in an anonymous repository.

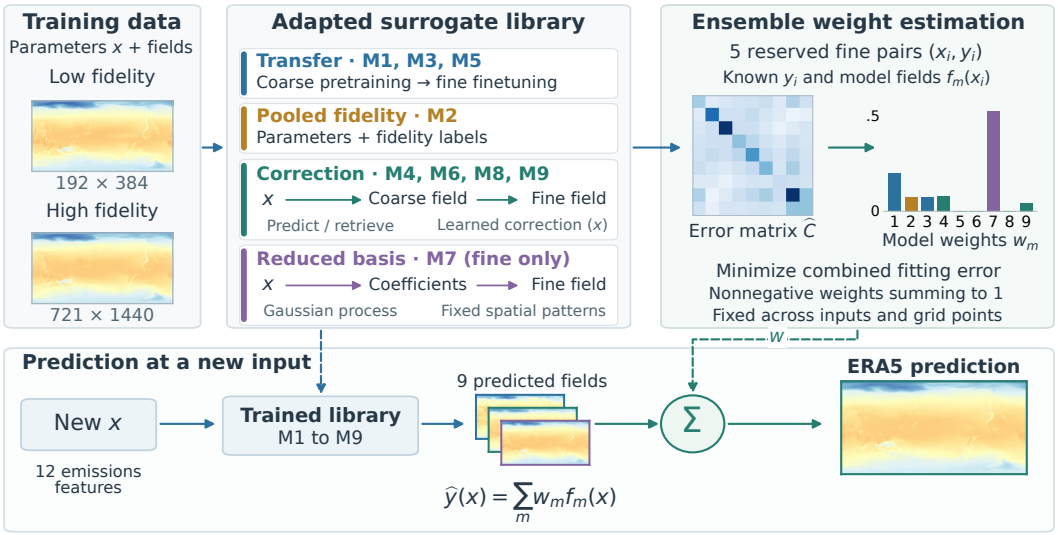


Figure 1: **AutoMF surrogate training, ensemble fitting, and prediction.** M1 to M9 are individually adapted models in our multifidelity surrogate library. Correctors predict or retrieve coarse fields and learn corrections. Proper orthogonal decomposition (POD) extracts spatial patterns, while Gaussian processes (GP) predict their coefficients. Five fine fitting examples determine the mixture weights. ERA5 uses 12 emissions features (Niu et al., 2024, Hersbach et al., 2020).

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.

## 1 INTRODUCTION

Scientific design requires solution fields at many physical parameter values. A surrogate predicts these fields without repeating fine simulations. Multifidelity learning uses low fidelity (LF) solutions alongside high fidelity (HF) data (Kennedy & O'Hagan, 2000, Howard et al., 2023, Niu et al., 2024), introducing a practical choice: how should a model use the coarse information? Transfer learning, joint training, and correction exploit different fidelity relationships, and their accuracy changes as those relationships weaken (Faza et al., 2026).

We study this choice through nine surrogates that we adapt from established architectures and reduced basis methods. Our adaptations specify parameter conditioning, fidelity supervision, and internal coarse reference construction. The library includes a fine only reference. Seven models lead within it on at least one PDE task, but the best model for a new input is unknown. Selecting one from a small fitting set can discard useful predictions. AUTOMF instead uses those same reserved fine examples to weight the library's predictions. Each dataset receives one weight per model, shared across inputs and grid points. Our central question is whether this combination reduces error from uncertain model choice within a fixed trained library.

Generating fine field datasets can require substantial simulation effort (Sung et al., 2025b,a). For example, the Regional Aerosol Model Intercomparison Project (RAMIP) reports approximately 100,000 core hours for CESM2.1 and 400,000 for UKESM1 per coupled climate simulation spanning roughly 36 simulated years (Wilcox et al., 2023, Appendix A). These external examples motivate reusing costly field data across surrogate candidates. Once trained, the library can be combined without retraining its members when better predictions justify the additional surrogate computation.

Our contributions are an adapted surrogate library, a 22 dataset benchmark, and an empirical evaluation of automated model choice. The benchmark combines our simulations with imported data and common evaluation cases within each task. The workflow compares selection and ensemble fitting using the same additional fine solutions. It builds on engineering surrogate ensembles (Goel et al., 2007, Acar & Rais-Rohani, 2009, Viana et al., 2009), automated model libraries (Feurer et al., 2015, Erickson et al., 2020), and comparisons of multifidelity field surrogates (Brunel et al., 2025), applying established combination principles to the adapted library. The contribution lies in this integration and its evaluation rather than a new weighting principle. With five fitting examples, the fitted mixture improves on selection on most datasets. Existing subset and fitting budget controls examine how much of that benefit simpler combinations recover.

## 2 BACKGROUND: SURROGATE SELECTION AND FIELD ENSEMBLES

Engineering surrogate ensembles use error based weighting (Goel et al., 2007), joint optimization of metamodel weights (Acar & Rais-Rohani, 2009, Strömberg, 2021), and comparisons of selection, weighting, and library subsets (Viana et al., 2009). For field outputs, Brunel et al. (2025) compare more than a dozen multifidelity surrogates and relate performance to fidelity correlation, training size, and residual complexity. We study selection and combination of adapted field surrogates under explicitly counted budgets of additional fine solutions.

Our weighting rules follow stacked generalization (Wolpert, 1992), stacked regression (Breiman, 1996), and Super Learner (van der Laan et al., 2007). Greedy ensemble selection (Caruana et al., 2004), auto sklearn (Feurer et al., 2015, 2022), and AutoGluon (Erickson et al., 2020) combine broader model libraries. Related multifidelity workflows automate hierarchical fusion of LF models (Wang et al., 2024), screening of LF information and surrogates (Lee et al., 2026), or family selection through leave one out validation (Kim et al., 2026). Our fitting stage combines completed field predictions from different fidelity strategies under a common parameter input interface.

The library adapts FNO, DeepONet, WNO, and Transolver components (Li et al., 2021, Lu et al., 2021, Tripura & Chakraborty, 2022, Wu et al., 2024). Other multifidelity methods include DeepONets (Howard et al., 2023), MF WNO (Thakur et al., 2022), FNO for carbon storage (Tang et al., 2023), MFRNP residual prediction (Niu et al., 2024), and FIRE distribution conditioning (Yu et al., 2026). Appendix B.1 distinguishes our changes from these sources. HyperNOs automates operator training (Ghiotto, 2026), and LANE SI combines sea ice fields (Borisova & Nikitin, 2024).

PDEBench and The Well provide varied physics simulations (Takamoto et al., 2022, Ohana et al., 2024). Our benchmark organizes parameter inputs and fields at several fidelities, with generated and imported data described in Appendix A. Fitting uses supplied data, leaving acquisition to methods such as active multi resolution learning (Li et al., 2024).

# 3 AUTOMF: MODEL SELECTION AND ENSEMBLE FITTING

## 3.1 PROBLEM FORMULATION AND DATA SPLITS

At fidelity $\ell$, the training data are $\mathcal{D}_\ell = \{(x_i^{(\ell)}, y_i^{(\ell)})\}_{i=1}^{N_\ell}$: $N_\ell$ examples indexed by $i$, each pairing $d$ simulation parameters $x_i^{(\ell)} \in \mathbb{R}^d$ with a field $y_i^{(\ell)}$ on grid $\Omega_\ell$. The index $\ell$ identifies fidelity. Each surrogate model $f_m$ maps the parameter vector $x$ to a fine field on a common target grid. Among models that use a coarse reference, some predict it at $x$ using a low fidelity surrogate, while others retrieve the coarse training field with the nearest parameters in standardized parameter space. The coarse reference is interpolated onto the target grid as needed and used in a learned correction to predict the fine field. The notation $f_m$ includes these steps, so prediction requires only $x$ and no new simulation.

A case is one parameter vector paired with its complete fine field. Reserved cases are divided into a *fitting set*, used to select a model or determine weights, and an *evaluation set*, used to score the resulting prediction. $K$ counts fitting cases. Both sets are excluded from low fidelity (LF) and high fidelity (HF) training before preprocessing or model fitting. Input identities establish correspondence between fidelities. Each dataset has its own trained library, held fixed across split assignments and fitting budgets. We save each rule's fitted choice before evaluation. Appendix A.2 explains the split procedures and historical evaluation access.

## 3.2 ADAPTED SURROGATE MODEL LIBRARY

M1 to M9 are our adapted surrogate models for multifidelity field prediction (Table 2). We construct them from established model components for a common parameter to field task, with changes to parameter conditioning, fidelity supervision, and coarse reference construction. Their field representations and uses of coarse data vary across the library. Transfer can reuse features across fidelities, joint supervision shares information during training, and correction uses a coarse field as a spatial reference. A fine only reduced basis model supplies an alternative when coarse information is less useful. We adapt these strategies to the same parameter input and target field interface, including the internal coarse prediction or retrieval required by correctors. Three models first learn from coarse fields and then finetune on fine fields, following multifidelity transfer learning (Lyu et al., 2023). FiLM FNO transfer combines Fourier mixing (Li et al., 2021) with parameter conditioned featurewise linear modulation (FiLM) (Perez et al., 2018). ConvNeXt transfer uses convolutional blocks in an encoder and decoder with skip connections (Liu et al., 2022). Wavelet transfer uses a custom wavelet representation inspired by WNO (Tripura & Chakraborty, 2022). Together they test coarse initialization across different spatial representations. All pairs FNO instead pools supervision across fidelity pairs before fine training. It takes parameters and two fidelity labels as inputs. Pair enumeration is inspired by Poseidon (Herde et al., 2024) and adapted here to fidelity labels and FNO training (Li et al., 2021).

Four models use coarse predictions or retrieved fields to guide a correction. Distribution FNO predicts a residual from coarse ensemble summaries, adapting FIRE's distribution conditioning idea (Yu et al., 2026). Retrieved field DeepONet uses a branch and trunk representation (Lu et al., 2021, 2022) to correct a neighboring coarse training field. The slice attention and ConvNeXt correctors refine a predicted coarse field through six updates, using modified Transolver inspired attention (Wu et al., 2024) and convolutional blocks (Liu et al., 2022), respectively. Their training coarse inputs are assembled out of fold with four ensemble members per case. Finally, POD GP predicts coefficients in a proper orthogonal decomposition (POD) basis using Gaussian processes (GP) (Higdon et al., 2008). It tests whether a compact field representation can be learned directly from fine solutions. Appendix B.1 explains for each model the source components, our changes, and the inclusion rationale. The resulting comparison evaluates complete surrogate models and training procedures.

### 3.3 Model selection and ensemble weighting

The ensemble averages the model predictions,

$$\widehat{y}_w(x) = \sum_{m=1}^{M} w_m f_m(x), \qquad w \in \Delta_M = \{w \in \mathbb{R}^M : w_m \geq 0,\ \mathbf{1}^\top w = 1\}. \tag{1}$$

Here $M = 9$ counts models, $w_m$ weights model $m$, and $\widehat{y}_w$ is the combined field. The simplex $\Delta_M$ contains nonnegative weights summing to one. The vector $\mathbf{1}$ contains ones and $\top$ denotes transpose. Each dataset's fitting set determines one weight vector. It stays fixed across new inputs and all grid points in the separate evaluation set. For fitting pair $(x_i, y_i)$, define $\langle a, b\rangle_Q = a^\top Q b$, $\|a\|_Q = \sqrt{a^\top Q a}$, and $s_i = \max(\|y_i\|_Q, 10^{-8})$. Fields are vectors of grid values, $Q = I$ is the identity matrix, and the floor prevents division by zero. We compute

$$e_{im} = \frac{f_m(x_i) - y_i}{s_i}, \qquad [G_i]_{mn} = \langle e_{im}, e_{in}\rangle_Q, \qquad \widehat{C} = \frac{1}{K}\sum_{i=1}^{K} G_i. \tag{2}$$

Here $e_{im}$ is model $m$'s relative error field on example $i$, $G_i$ records error products between models $m$ and $n$, and $\widehat{C}$ averages these over $K$ fields. Diagonal entries measure individual squared error. Other entries measure error alignment, retaining information needed to score a mixture. To determine the final ensemble composition, we evaluate three distinct weighting strategies using the reserved fitting examples:

1. **Selected model.** This rule gives weight one to model $m^*$ with the lowest mean relative $L^2$ error on the fitting examples, $m^* = \arg\min_m K^{-1}\sum_i \sqrt{[G_i]_{mm}}$.
2. **Inverse error mixture.** This rule computes weights directly from individual mean squared relative errors, giving more accurate models more weight:

$$w_m^{\text{inv}} = \frac{1/\max(\widehat{C}_{mm}, \tau)}{\sum_j 1/\max(\widehat{C}_{jj}, \tau)}, \qquad \tau = 10^{-30}. \tag{3}$$

   Here $w_m^{\text{inv}}$ is the normalized inverse error weight, $j$ runs over all models, and $\tau$ prevents division by zero. The weights sum to one.
3. **Fitted mixture.** This rule chooses the weights together by minimizing the error of their combined prediction:

$$\widehat{w}^{\text{full}} \in \arg\min_{w \in \Delta_M} w^\top \widehat{C} w. \tag{4}$$

Here $\widehat{w}^{\text{full}}$ minimizes the mean squared relative error $w^\top \widehat{C} w$ on the fitting examples. This is convex stacking (Wolpert, 1992, Breiman, 1996), allowing a less accurate model to offset another's errors. Both mixture rules use squared normalized errors, whereas model selection and reporting use mean relative $L^2$. The objectives can have different optima. We test the consequence of this choice in Section 5 and derive the distinction in Appendix F.1. Numerical details appear in Appendix C.1.

Ensemble fitting estimates only nine weights, with eight free values because they sum to one. The surrogate models remain fixed, and each fitting field supplies spatial error products that summarize individual accuracy and error alignment. Once these products are computed, the optimization uses a $9 \times 9$ matrix regardless of the number of grid points. This small fitting problem makes limited additional fine data a practical option. Spatial values within a field remain dependent, however, so field resolution does not count as additional independent fitting examples or guarantee generalization from five fields. Appendix C.2 gives the pseudocode.

## 4 Experimental setup

### 4.1 Benchmark datasets and comparison scope

The benchmark contains 21 PDE tasks in seven classes: elliptic equations, diffusion, convection, shocks, porous flow, reaction diffusion, and waves. ERA5 supplies the 22nd task in climate

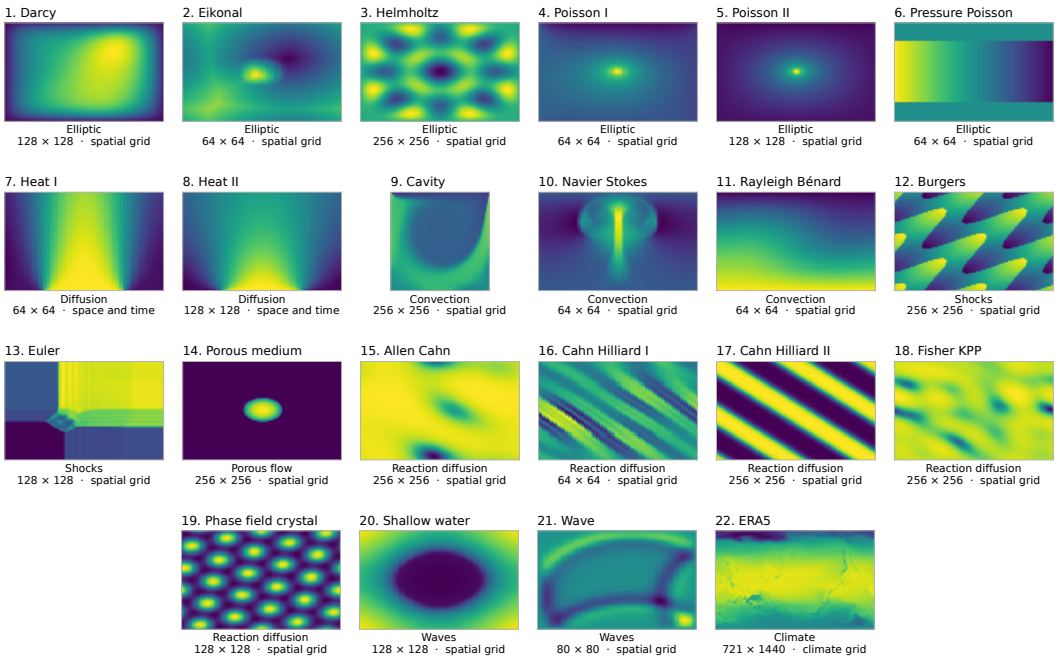


Figure 2: **The 22 datasets span different field structures.** Panels are grouped by class and show one fine training field per dataset with its native grid size. Heat fields span space and time, and the cavity shows vorticity. Display conventions are described in Appendix A.3. Imported data: Poisson II, Heat II, and Navier Stokes (Niu et al., 2024), and ERA5 (Hersbach et al., 2020, Niu et al., 2024).

emulation (Hersbach et al., 2020, Niu et al., 2024). Figure 2 shows one fine field from each dataset, illustrating the range of field structures and grid resolutions. The collection combines our simulations with imported parameter and field data, using different grids or data sources as fidelities. Euler, Burgers, and Shallow water use the published PyClaw finite volume framework (Ketcheson et al., 2012). Their problem designs include Euler quadrant configurations motivated by The Well (Ohana et al., 2024) and a radial dam break adapted from PDEBench (Takamoto et al., 2022). Custom solvers are supported by targeted analytical and numerical checks, including a Barenblatt solution test for Porous medium and mesh refinement for Allen Cahn (Appendix A.1). Tasks can share an equation's mathematical form while varying its coefficients, initial or boundary conditions, or sampled parameter ranges. For example, the heat equation describes different temperature fields as diffusivity and boundary fluxes change. The roster therefore counts prediction tasks, not independent physical families (Appendix A).

All nine library models and their ensembles cover all 22 datasets. Comparisons use the same evaluation cases within each dataset and split. Final surrogates use a single seed and the training procedures documented in Appendices B and D.1. Comparisons between the three ensemble rules hold both the trained models and fitting examples fixed.

### 4.2 Fitting budgets and evaluation metrics

For each PDE dataset, we use three distinct splits to divide the reserved cases into fitting and evaluation sets, keeping the trained models fixed. To test varying data allowances, these fitting sets are strictly nested: the set of five cases is contained in the set of ten, which is contained in the set of twenty. Thus $K \in \{5, 10, 20\}$ counts fitting examples, while the three splits vary which examples are used. Every rule receives the same fitting cases, and the evaluation cases stay fixed across all three budgets within each split. ERA5 instead uses three five case subsets of a ten case fitting pool at $K = 5$, all ten cases at $K = 10$, and the same seven evaluation cases throughout. These repetitions measure sensitivity to fitting choices, not training seeds (Appendix A.2).

We average relative $L^2$ errors over evaluation cases within each split, then average the three split scores. To compare proportional changes across datasets, we aggregate error ratios geometrically. Dataset aggregation gives each dataset equal weight. PDE class aggregation first averages within each class and then gives the classes equal weight:

$$\rho_{a/b} = \exp\left[\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{|\mathcal{D}_c|}\sum_{d\in\mathcal{D}_c}\log\frac{\bar{E}_{d,a}}{\bar{E}_{d,b}}\right]. \tag{5}$$

Here $a, b$ denote methods, $\bar{E}_{d,a}$ is method $a$'s partition averaged error on dataset $d$, $\mathcal{C}$ is the set of PDE classes, and $\mathcal{D}_c$ contains class $c$'s datasets. Vertical bars count set members. The ratio $\rho_{a/b}$ is a weighted geometric mean of the per dataset ratios $\bar{E}_{d,a}/\bar{E}_{d,b}$, with $\rho_{a/b} < 1$ favoring $a$. PDE class aggregation gives each class equal total weight, divided equally among its datasets. Dataset aggregation gives each task equal weight and includes ERA5. These reporting weights are distinct from the model weights used to predict a field. A win means lower error than the comparison method, and a loss means higher error, using unrounded errors averaged across partitions. Equal errors are ties.

# 5 Results

## 5.1 Individual surrogate performance across datasets

Seven of the nine members have the lowest error within the library on at least one PDE dataset. Wavelet transfer leads on Darcy, POD GP on Poisson I and II, distribution FNO on the heat datasets, and the ConvNeXt corrector on Cavity and Euler. These models use different field representations and different ways of learning from the available fidelities. Table 1 shows their individual errors beside the three ensemble rules and the best implemented baseline. The variation supports retaining several adapted surrogates rather than choosing one architecture for the entire benchmark.

The library also contains strong individual predictions relative to the baselines. Its best member has lower error than the best implemented baseline on 16 of the 21 PDE datasets. Both best members in this comparison are identified from evaluation errors, separately for each dataset. The selected model rule instead uses only the fitting examples to choose one of M1 to M9 before evaluation. This separates available accuracy from selection using reserved examples.

## 5.2 Ensemble comparison with individual surrogates

With five fitting examples, the fitted mixture achieves lower error than the selected model on 17 of 22 datasets. The reduction in geometric mean relative $L^2$ error is 11.6% when the seven PDE classes receive equal weight over the 21 PDE tasks, and 7.5% when all 22 datasets, including ERA5, receive equal weight. Both rules use the same trained models and fitting examples, so this comparison measures the benefit of combining their predictions within the fixed library.

Both mixtures outperform every individual library member on Heat I, Heat II, and Cavity (Table 1), showing that combination can improve on an already accurate model. Figure 3 illustrates this behavior on one Heat I input. Across the 21 PDE tasks, the fitted mixture's error ratios against the retrospectively best library member are 0.923 under PDE class weighting and 0.989 under dataset weighting (Appendix D.2). The reference selects one member per dataset after averaging partitions. The smaller gain against this reference indicates that much of the benefit comes from reducing errors in model choice. Combination can exceed the best individual prediction, but does not do so consistently. Both mixtures increase error on Poisson I and II.

## 5.3 Comparison with implemented baselines

The fitted mixture reduces geometric mean error relative to the best implemented baseline by 29.4% under equal weighting of the seven PDE classes and 17.5% under dataset weighting on the same 21 PDE tasks (Appendix D.2). The baseline is chosen separately for each task using evaluation error. The mixture has lower error on 14 tasks and higher error on seven. The fitted mixture improves on selection in six of the seven PDE classes and on the best implemented baseline in five. The largest baseline gains occur in Diffusion and Porous flow. Appendix E.2 reports each class and the

Table 1: **Comparing the library, its combinations, and baselines.** All columns report mean relative $L^2$ error in percent on the same evaluation cases. M1 to M9 are the adapted library models. The best baseline is the lowest evaluation error among the implemented references B1 to B11 in each row, with its B label identifying the entry. Table 2 gives model names and aggregate comparisons. The selected model and both mixtures use five fitting examples. Bold marks the lowest unrounded evaluation error. Imported data: Poisson II, Heat II, and Navier Stokes (Niu et al., 2024), and ERA5 (Hersbach et al., 2020, Niu et al., 2024).

| Dataset | Best baseline | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | Selected model | Inverse error mixture | Fitted mixture |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Darcy | 3.37 (B1) | 3.07 | 3.44 | 2.78 | 4.98 | **2.32** | 31.7 | 3.37 | 3.41 | 3.41 | 2.52 | 2.55 | 2.33 |
| Eikonal | 2.21 (B8) | 2.42 | 2.15 | 1.81 | 4.79 | 2.59 | 7.37 | 3.63 | 3.74 | 2.95 | 1.81 | 1.76 | **1.76** |
| Helmholtz | 4.12 (B9) | **3.3** | 3.48 | 4.79 | 10.9 | 5.04 | 29.1 | 7.1 | 10.9 | 10.8 | 6.25 | 6.89 | 6.55 |
| Poisson I | **0.00531** (B2) | 0.0413 | 0.0387 | 0.0427 | 0.0611 | 0.0955 | 9.92 | 0.00684 | 0.128 | 0.0838 | 0.00684 | 0.008 | 0.00788 |
| Poisson II | 0.0128 (B1) | 0.256 | 0.536 | 0.2 | 25.5 | 0.881 | 26.6 | **0.0121** | 0.921 | 0.798 | **0.0121** | 0.0125 | 0.0129 |
| Pressure Poisson | 0.0639 (B9) | 0.0599 | 0.0695 | 0.0918 | 0.0871 | **0.0498** | 0.285 | 0.0922 | 0.309 | 0.145 | 0.0734 | 0.0769 | 0.0802 |
| Heat I | 0.0206 (B10) | 0.0164 | 0.0125 | 0.0105 | 0.00915 | 0.0268 | 2.28 | 0.327 | 0.0494 | 0.0149 | 0.00972 | **0.00593** | 0.0062 |
| Heat II | 0.0201 (B10) | 0.0139 | 0.00878 | 0.00859 | 0.00662 | 0.0173 | 1.58 | 0.312 | 0.0268 | 0.0121 | 0.00662 | 0.00437 | **0.00435** |
| Cavity | **0.0419** (B5) | 98.5 | 0.29 | 98.5 | 0.681 | 3.6 | 70.1 | 0.409 | 2.06 | 0.164 | 0.164 | 0.142 | 0.142 |
| Navier Stokes | 1.52 (B10) | 1.55 | 1.69 | 1.57 | 1.58 | 2.33 | 4.41 | 2.79 | 2.62 | 1.95 | 1.61 | 1.52 | **1.5** |
| Rayleigh Bénard | 0.011 (B10) | 0.0112 | 0.00915 | 0.00715 | 0.00301 | 0.0137 | 0.0999 | 0.295 | 0.0529 | 0.0195 | 0.00301 | 0.00286 | **0.00269** |
| Burgers | 0.562 (B9) | 0.547 | 0.533 | 100 | 0.507 | 1.28 | 4.88 | 2.5 | 1.69 | 0.545 | 0.516 | 0.513 | **0.502** |
| Euler | **5.54** (B2) | 6.2 | 6.08 | 6.72 | 6.3 | 7.43 | 8.13 | 6.15 | 6 | 5.88 | 6.09 | 5.64 | 5.73 |
| Porous medium | 0.0718 (B9) | 0.0448 | 0.0325 | 0.0596 | 0.0654 | 0.126 | 2.75 | 0.909 | 0.681 | 0.0497 | 0.0325 | **0.0262** | 0.0263 |
| Allen Cahn | 41.2 (B5) | 53 | 52.7 | 42.1 | 45.4 | 41.3 | 102 | 55.5 | 41.4 | 38.8 | 39.4 | 37.9 | **34.6** |
| Cahn Hilliard I | 13.3 (B10) | 12.9 | 12.5 | 12.2 | 11.7 | 36.3 | 81.6 | 30.3 | 41.1 | 12.5 | 12.2 | 11.3 | **10.7** |
| Cahn Hilliard II | 49.8 (B5) | **46.9** | 48.3 | 51.9 | 50.8 | 60.8 | 99.8 | 64.8 | 55.9 | 47.8 | 52.2 | 47.2 | 49.3 |
| Fisher KPP | 3.24 (B9) | 3.21 | 3.22 | 2.36 | 2.98 | 3.83 | 6.49 | 6.86 | 2.86 | 2.84 | 2.36 | 2.53 | **2.3** |
| Phase field crystal | 83.5 (B11) | 89.6 | 89.5 | 111 | 83.3 | 202 | 120 | 83.5 | 85.3 | 85.7 | 83.5 | 83.8 | **83.3** |
| Shallow water | 0.403 (B10) | 0.428 | 0.367 | 0.362 | 0.876 | 0.446 | 1.38 | 1.09 | 1.29 | 0.502 | 0.367 | **0.357** | 0.389 |
| Wave | **3.3** (B3) | 6.36 | 6.31 | 4.75 | 11.1 | 5.04 | 64.2 | 24.9 | 8.35 | 11.3 | 4.84 | 4.39 | 4.27 |
| ERA5 | **5** (B1) | 14.2 | 17.7 | 20.4 | 12.2 | 9.75 | 9.54 | 6.74 | 20.4 | 8.24 | 6.74 | 5.81 | 6.49 |

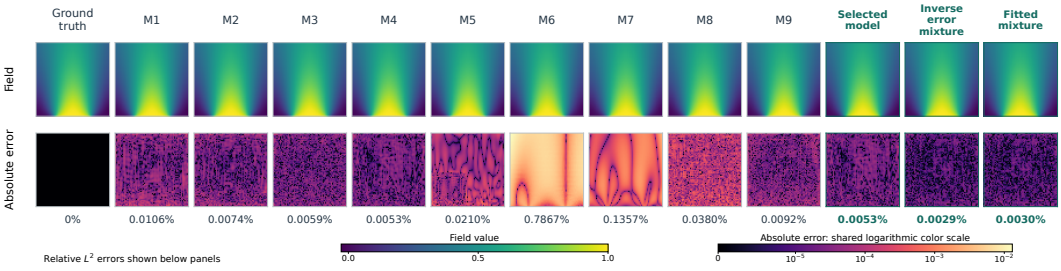


Figure 3: **Combining models reduces field error on a Heat I example.** Top: ground truth, nine model predictions, and the selected model, inverse error mixture, and fitted mixture. Bottom: absolute error on a shared logarithmic color scale, with relative $L^2$ error below each panel. Fields share a linear color scale and a $64 \times 64$ space and time grid. The three rules use five fitting examples. The selected model is M4 here. The fitted mixture lowers error by 44.4% relative to the best individual model.

sensitivity of the aggregate to their inclusion. Table 2 and Appendix D.1 report the implemented comparisons, with adaptations in Appendix B.1 and ERA5 results in Section 7.

### 5.4 Comparison of ensemble weighting rules

The inverse error mixture assigns weights from individual fitting errors, while the fitted mixture minimizes the error of the combined prediction. They occupy the top two Elo positions (Table 2), and joint fitting provides only a small further reduction in aggregate error under either reported aggregation. Individual fitting errors therefore recover most of the gain. The relative ordering varies across tasks: joint fitting improves on inverse weighting for Allen Cahn, while inverse weighting is more accurate on ERA5.

Table 2: **Model sources and performance, ranked by Elo.** Each ratio divides a method's mean relative $L^2$ evaluation error by the Selected model's error, then aggregates geometrically. Selection chooses the lowest error model among M1 to M9 on the five fitting examples. **PDE class ratio** averages within each class and equally across the seven PDE classes (21 datasets). **Dataset ratio** weights all 22 datasets equally, including ERA5. Lower is better, with 0.90 meaning 10% lower aggregate error than selection. Higher Elo is better (Appendix D.5). M1 to M9 are our adapted models and B1 to B11 are baseline entries. Citations identify components or ideas, with adaptations in Appendix B.1. Individual errors appear in Table 1 and Appendix D.1. Training costs and fine data budgets differ.

| Model / rule | Component or idea source | PDE class ratio | Dataset ratio | Elo |
|---|---|---|---|---|
| Fitted mixture | (Breiman, 1996) | 0.884 | 0.925 | 1988 |
| Inverse error mixture | (Goel et al., 2007) | 0.888 | 0.932 | 1951 |
| Selected model | Fitting set selection | 1.000 | 1.000 | 1851 |
| M2 All pairs FNO | (Li et al., 2021, Herde et al., 2024) | 1.333 | 1.541 | 1688 |
| M1 FiLM FNO transfer | (Li et al., 2021, Perez et al., 2018) | 1.940 | 2.035 | 1642 |
| M9 ConvNeXt corrector | (Liu et al., 2022) | 1.695 | 1.867 | 1634 |
| M3 ConvNeXt transfer | (Liu et al., 2022) | 2.607 | 2.445 | 1619 |
| M4 Distribution FNO | (Yu et al., 2026) | 1.795 | 2.155 | 1592 |
| B10 FNO transfer II | (Lyu et al., 2023) | 1.794 | 1.850 | 1590 |
| B9 FNO transfer I | (Lyu et al., 2023) | 1.794 | 1.848 | 1578 |
| M5 Wavelet transfer | (Tripura & Chakraborty, 2022) | 2.459 | 2.400 | 1528 |
| B5 Latent MF network | (Li et al., 2022a) | 3.794 | 2.772 | 1497 |
| M7 POD GP | (Higdon et al., 2008) | 5.492 | 3.108 | 1450 |
| B1 Autoregressive POD GP | (Kennedy & O'Hagan, 2000) | 5.854 | 3.362 | 1449 |
| M8 Slice attention corrector | (Wu et al., 2024) | 4.189 | 3.549 | 1439 |
| B2 Nonlinear POD GP | (Perdikaris et al., 2017) | 5.871 | 3.626 | 1424 |
| B3 Composite MF MLP | (Meng & Karniadakis, 2020) | 4.174 | 3.974 | 1387 |
| B8 Fidelity basis FNO | (Li et al., 2022b) | 8.979 | 6.217 | 1364 |
| B4 Composite MF DeepONet | (Howard et al., 2023) | 8.001 | 5.803 | 1337 |
| B11 Affine POD GP | (Zhang et al., 2018) | 17.624 | 11.236 | 1218 |
| B6 Nonlinear decoder transfer | (Seidman et al., 2022) | 9.927 | 8.252 | 1197 |
| M6 Retrieved field DeepONet | (Lu et al., 2022) | 20.847 | 14.890 | 1057 |
| B7 MFRNP adapter | (Niu et al., 2024) | 49.969 | 24.194 | 1020 |

Uniform averaging loses to selection on 19 of the 21 PDE datasets, but can be hurt by individually poor models. A stronger control averages the three models with lowest fitting error. On the 17 task diagnostic subset, its class error ratio against selection is 0.997, compared with 0.876 for the full fitted mixture. Fitting weights within those three models gives 0.889, recovering much of the full mixture's benefit (Appendix D.4). These results support weighting accurate candidates, without establishing that all nine predictions are necessary for every task. Error complementarity can benefit both weighting rules (Appendix F.1).

### 5.5 Sensitivity to fitting budget and dataset choice

Increasing the fitting allowance from five to twenty examples raises the fitted mixture's reduction over selection from 11.6% to 13.1% under PDE class weighting (Appendix D.3). Models and evaluation cases remain fixed within each split, and both rules receive the same allowance. Five examples yield useful weights here, without establishing a universally sufficient fitting budget.

The supplementary subset checks cover the original 17 PDE tasks and retain gains over selection across the reported exclusions (Appendix E.1). Comparisons against the retrospectively best individual member are more sensitive to which tasks are included. A separate loss control on those 17 tasks modestly improves accuracy by fitting mean relative $L^2$ error directly rather than its square (Table 15). These diagnostics reuse saved predictions and cover a subset of the benchmark. Separately, Appendix E.3 reports an existing training budget sweep comparing coarse pretraining

with fine only training for one surrogate architecture. It provides supplementary evidence about transfer, distinct from fitting the complete AutoMF library.

### 5.6 Separate evaluation on new Heat I inputs

We test the existing Heat I library on 128 newly generated evaluation inputs, disjoint from all original training and evaluation inputs. A separate pool of 40 new cases supplies the fitting examples. With five fitting examples, the inverse error and fitted mixtures reduce error relative to model selection by 38.6% and 36.9%, respectively. Neural checkpoints remain fixed and weights are saved before evaluation answers are scored. This separate case study supports a benefit on new inputs of the same PDE and is excluded from the 22 dataset aggregate. Table 14 reports all budgets.

## 6 Ensemble error analysis

For a fixed library, the population squared relative error is

$$R(w) = \mathbb{E}[w^\top G(x,y)w] = w^\top C w, \qquad C = \mathbb{E}[G(x,y)]. \tag{6}$$

Here $G(x,y)$ contains error products for an input and target pair, $\mathbb{E}$ averages over pairs, and $C$ is the population mean. Cross terms retain error alignment and shared bias (Breiman, 1996, Krogh & Vedelsby, 1994), allowing either mixture to benefit from partly offsetting errors. The reported unsquared metric $L(w) = \mathbb{E}[\sqrt{w^\top G w}]$ can favor different weights from $R$. Appendix F.1 gives a counterexample and direct fitting control. Neither this identity nor the bound in Appendix F.2 guarantees accurate weights from five fields.

## 7 ERA5 temperature field emulation

We use the emissions to temperature dataset released by Niu et al. (2024). Its twelve inputs comprise $CO_2$ and $CH_4$ emissions and five principal component scores for each of black carbon and $SO_2$, following ClimateBench (Watson-Parris et al., 2022). Thirteen ScenarioMIP climate simulators supply coarse fields at eight resolutions. ERA5 provides global air temperature two metres above the surface on the released $721 \times 1440$ grid (Hersbach et al., 2020, Niu et al., 2024). We use 55 fine inputs for training, ten for the fitting pool, and seven for evaluation, excluding reserved inputs from every training fidelity. Appendix E.4 describes the source protocol, verified case correspondence, and the scoring convention.

POD GP is the most accurate library member at 6.7369% mean relative $L^2$ error. With five fitting examples, the inverse error mixture reaches 5.8146% and the fitted mixture reaches 6.4877%, reductions of 13.69% and 3.70% from POD GP. Both improve on the best library member, although some baselines remain more accurate (Table 1 and Appendix D.1).

## 8 Discussion and limitations

Combining a fixed library reduces error from uncertain model choice. Inverse error weighting and smaller fitted subsets recover much of the gain. Training several models adds computation, and net savings relative to simulation cost remain unmeasured.

The main comparison uses one final training seed per surrogate. Repeated fitting splits vary reserved examples, not trained models. Historical evaluation informed development. The new Heat I case study tests one PDE (Section 5.6), and the transfer sweep studies one architecture, leaving coarse data's benefit for the full ensemble unresolved.

## 9 Conclusion

AutoMF combines nine adapted surrogates using five additional fine examples, without a new coarse simulation at deployment. The fitted mixture improves on selection on 17 of 22 datasets, and both mixtures improve on new Heat I inputs. Inverse error weighting recovers most of the aggregate gain. These results support combining available predictions when model choice is uncertain.

## REFERENCES


E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37:279–294, 2009. doi: 10.1007/s00158-008-0230-y. URL https://doi.org/10.1007/s00158-008-0230-y.

Erdem Acar. Effect of error metrics on optimum weight factor selection for ensemble of metamodels. *Expert Systems with Applications*, 42(5):2703–2709, 2015. doi: 10.1016/j.eswa.2014.11.020. URL https://doi.org/10.1016/j.eswa.2014.11.020.

Julia Borisova and Nikolay O. Nikitin. Surrogate modelling for sea ice concentration using lightweight neural ensemble. In *AAAI Workshop on AI to Accelerate Science and Engineering*, 2024. URL https://arxiv.org/abs/2312.04330. Preprint arXiv:2312.04330, 2023.

Leo Breiman. Stacked regressions. *Machine Learning*, 24:49–64, 1996. doi: 10.1023/A:1018046112532. URL https://statistics.berkeley.edu/sites/default/files/tech-reports/367.pdf.

Lucas Brunel, Mathieu Balesdent, Loïc Brévault, Rodolphe Le Riche, and Bruno Sudret. A survey on multi-fidelity surrogates for simulators with functional outputs: unified framework and benchmark. *Computer Methods in Applied Mechanics and Engineering*, 435:117577, 2025. doi: 10.1016/j.cma.2024.117577. URL https://doi.org/10.1016/j.cma.2024.117577.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning*, 2004. doi: 10.1145/1015330.1015432. URL https://doi.org/10.1145/1015330.1015432.

K. R. Elder and Martin Grant. Modeling elastic and plastic deformations in nonequilibrium processing using phase field crystals. *Physical Review E*, 70:051605, 2004. doi: 10.1103/PhysRevE.70.051605. URL https://doi.org/10.1103/PhysRevE.70.051605.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. In *7th ICML Workshop on Automated Machine Learning*, 2020. URL https://arxiv.org/abs/2003.06505.

Ghifari Adam Faza, Kemas Zakaria, Pramudita Satria Palar, Keivan Shariatmadar, Lavi Rizki Zuhal, Hans Hallez, and David Moens. Benchmarking multi-fidelity neural operators on complex PDE problems with non-trivial fidelity differences, 2026. URL https://arxiv.org/abs/2608.04708.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/11d0e6287202fced83f79975ec59a3a6-Abstract.html.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-Sklearn 2.0: Hands-free AutoML via meta-learning. *Journal of Machine Learning Research*, 23, 2022. URL https://jmlr.org/papers/v23/21-0992.html.

Massimiliano Ghiotto. HyperNOs: automated and parallel library for neural operators research. *Bollettino dell'Unione Matematica Italiana*, 19:709–743, 2026. doi: 10.1007/s40574-025-00516-0. URL https://doi.org/10.1007/s40574-025-00516-0. First published online October 21, 2025.

Tushar Goel, Raphael T. Haftka, Wei Shyy, and Nestor V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33:199–216, 2007. doi: 10.1007/s00158-006-0051-9. URL https://doi.org/10.1007/s00158-006-0051-9.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/84e1b1ec17bb11c57234e96433022a9a-Abstract-Conference.html.

Hans Hersbach, Bill Bell, Paul Berrisford, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146:1999–2049, 2020. doi: 10.1002/qj.3803. URL https://doi.org/10.1002/qj.3803.

Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482): 570–583, 2008. doi: 10.1198/016214507000000888.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://doi.org/10.1080/01621459.1963.10500830.

Amanda A. Howard, Mauro Perego, George E. Karniadakis, and Panos Stinis. Multifidelity deep operator networks for data-driven and physics-informed problems. *Journal of Computational Physics*, pp. 112462, 2023. doi: 10.1016/j.jcp.2023.112462. URL https://arxiv.org/abs/2204.09157.

Marc C. Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000. doi: 10.1093/biomet/87.1.1. URL https://doi.org/10.1093/biomet/87.1.1.

David I. Ketcheson, Kyle T. Mandli, Aron J. Ahmadia, Amal Alghamdi, Manuel Quezada de Luna, Matteo Parsani, Matthew G. Knepley, and Matthew Emmett. PyClaw: Accessible, extensible, scalable tools for wave propagation problems. *SIAM Journal on Scientific Computing*, 34 (4):C210–C231, 2012. doi: 10.1137/110856976. URL https://doi.org/10.1137/110856976.

Jieon Kim, Taeyeon Park, Sanghyun Cho, Jieun Park, and Gunwoo Noh. MAESTRO: A multi-fidelity modeling assisted by an ensemble of surrogates for robust prediction. SSRN preprint, 2026. URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6207840. Posted February 10, 2026.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, 1994. URL https://proceedings.neurips.cc/paper/1994/hash/b8c37e33defde51cf91e1e03e51657da-Abstract.html.

Mingyu Lee, Juyoung Lee, Jae-Hoon Choi, Nam H. Kim, and Ikjin Lee. A novel adaptive quality-based multi-fidelity surrogate framework for multiple low-fidelity data sources. *Advanced Engineering Informatics*, 69:103973, 2026. doi: 10.1016/j.aei.2025.103973. URL https://web.mae.ufl.edu/nkim/Papers/paper151.pdf. Available online October 17, 2025.

Shibo Li, Zheng Wang, Robert M. Kirby, and Shandian Zhe. Deep multi-fidelity active learning of high-dimensional outputs. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 1694–1711, 2022a. URL https://proceedings.mlr.press/v151/li22b.html.

Shibo Li, Zheng Wang, Robert M. Kirby, and Shandian Zhe. Infinite-fidelity coregionalization for physical simulation. In *Advances in Neural Information Processing Systems*, volume 35, 2022b. URL https://arxiv.org/abs/2207.00678.

Shibo Li, Xin Yu, Wei Xing, Robert Kirby, Akil Narayan, and Shandian Zhe. Multi-resolution active learning of fourier neural operators. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2440–2448, 2024. URL https://proceedings.mlr.press/v238/li24k.html.

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://arxiv.org/abs/2010.08895.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.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.

Lu Lu, Raphaël Pestourie, Steven G. Johnson, and Giuseppe Romano. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Physical Review Research*, 4(2):023210, 2022. doi: 10.1103/PhysRevResearch.4.023210.

Yanfang Lyu, Xiaoyu Zhao, Zhiqiang Gong, Xiao Kang, and Wen Yao. Multi-fidelity prediction of fluid flow and temperature field based on transfer learning using Fourier Neural Operator, 2023. URL https://arxiv.org/abs/2304.06972.

Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 401:109020, 2020. doi: 10.1016/j.jcp.2019.109020.

Ruijia Niu, Dongxia Wu, Kai Kim, Yian Ma, Duncan Watson-Parris, and Rose Yu. Multi-fidelity residual neural processes for scalable surrogate modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38381–38394, 2024. URL https://proceedings.mlr.press/v235/niu24d.html.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, et al. The well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4f9a5acd91ac76569f2fe291b1f4772b-Abstract-Datasets_and_Benchmarks_Track.html.

Lauren Partin, Gianluca Geraci, Ahmad Rushdi, Michael S. Eldred, and Daniele E. Schiavazzi. Multifidelity data fusion in convolutional encoder/decoder networks. *Journal of Computational Physics*, 2022. doi: 10.1016/j.jcp.2022.111666. URL https://arxiv.org/abs/2205.05187.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D. Lawrence, and George E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A*, 473(2198):20160751, 2017. doi: 10.1098/rspa.2016.0751.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11671.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://arxiv.org/abs/1505.04597.

Jacob H. Seidman, Georgios Kissas, Paris Perdikaris, and George J. Pappas. NOMAD: Nonlinear manifold decoders for operator learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/24f49b2ad9fbe65eefbfd99d6f6c3fd2-Abstract-Conference.html.

Ira J. S. Shokar, Rich R. Kerswell, and Peter H. Haynes. Conditioning on PDE parameters to generalise deep learning emulation of stochastic and chaotic dynamics. arXiv preprint arXiv:2509.09599, 2025. URL https://arxiv.org/abs/2509.09599.

Niclas Strömberg. Comparison of optimal linear, affine and convex combinations of metamodels. *Engineering Optimization*, 53(4):702–718, 2021. doi: 10.1080/0305215X.2020.1746781. URL https://doi.org/10.1080/0305215X.2020.1746781.

Nicholas Sung, Steven Spreizer, Mohamed Elrefaie, Matthew C. Jones, and Faez Ahmed. BlendedNet++: A large-scale blended wing body aerodynamics dataset and benchmark, 2025a. URL https://arxiv.org/abs/2512.03280v1. arXiv:2512.03280v1.

Nicholas Sung, Steven Spreizer, Mohamed Elrefaie, Kaira Samuel, Matthew C. Jones, and Faez Ahmed. BlendedNet: A blended wing body aircraft dataset and surrogate model for aerodynamic predictions. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 3B: 51st Design Automation Conference, pp. V03BT03A049, 2025b. doi: 10.1115/DETC2025-168977. URL https://doi.org/10.1115/DETC2025-168977.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan 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.

Hewei Tang, Qingkai Kong, and Joseph P. Morris. Multi-fidelity fourier neural operator for fast modeling of large-scale geological carbon storage, 2023. URL https://arxiv.org/abs/2308.09113.

Akshay Thakur, Tapas Tripura, and Souvik Chakraborty. Multi-fidelity wavelet neural operator with application to uncertainty quantification, 2022. URL https://arxiv.org/abs/2208.05606.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations, 2022. URL https://arxiv.org/abs/2205.02191.

Mark J. van der Laan, Eric C. Polley, and Aaron E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1):Article 25, 2007. doi: 10.2202/1544-6115.1309. URL https://doi.org/10.2202/1544-6115.1309.

Juan Luis Vázquez. *The Porous Medium Equation: Mathematical Theory*. Oxford Mathematical Monographs. Oxford University Press, 2007. doi: 10.1093/acprof:oso/9780198569039.001.0001. URL https://academic.oup.com/book/7922.

Felipe A. C. Viana, Raphael T. Haftka, and Valder Steffen, Jr. Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization*, 39:439–457, 2009. doi: 10.1007/s00158-008-0338-0. URL https://doi.org/10.1007/s00158-008-0338-0.

Yitang Wang, Yong Pang, Tianhang Xue, Shuai Zhang, and Xueguan Song. Ensemble learning based hierarchical surrogate model for multi-fidelity information fusion. *Advanced Engineering Informatics*, 60:102535, 2024. doi: 10.1016/j.aei.2024.102535. URL https://doi.org/10.1016/j.aei.2024.102535.

D. Watson-Parris, Y. Rao, D. Olivié, Ø. Seland, P. Nowack, G. Camps-Valls, P. Stier, S. Bouabid, M. Dewey, E. Fons, J. Gonzalez, P. Harder, K. Jeggle, J. Lenhardt, P. Manshausen, M. Novitasari, L. Ricard, and C. Roesch. ClimateBench v1.0: A benchmark for data-driven climate projections. *Journal of Advances in Modeling Earth Systems*, 14:e2021MS002954, 2022. doi: 10.1029/2021MS002954. URL https://doi.org/10.1029/2021MS002954.

Laura J. Wilcox, Robert J. Allen, Bjørn H. Samset, et al. The regional aerosol model intercomparison project (RAMIP). *Geoscientific Model Development*, 16(15):4451–4479, 2023. doi: 10.5194/gmd-16-4451-2023. URL https://gmd.copernicus.org/articles/16/4451/2023/gmd-16-4451-2023.html.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1. URL https://doi.org/10.1016/S0893-6080(05)80023-1.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705, 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Rosen Ting-Ying Yu, Nicholas Sung, and Faez Ahmed. FIRE: Multi-fidelity regression with distribution-conditioned in-context learning using tabular foundation models, 2026. URL https://arxiv.org/abs/2601.22371. Also presented at the ICLR 2026 FM4Science workshop.

Yiming Zhang, Nam-Ho Kim, Chanyoung Park, and Raphael T. Haftka. Multifidelity surrogate based on single linear regression. *AIAA Journal*, 56(12):4944–4952, 2018. doi: 10.2514/1.J057299.

## TABLE OF CONTENTS FOR APPENDICES

**A Datasets and evaluation protocol** **16**
A.1 Dataset definitions . . . 16
A.2 Training, fitting, and evaluation cases . . . 17
A.3 Field visualization conventions . . . 17

**B Model adaptations and training** **18**
B.1 Model sources, adaptations, and rationale . . . 18
B.2 Training settings and data use . . . 20

**C Ensemble implementation** **22**
C.1 Numerical optimization and computational cost . . . 22
C.2 AutoMF training and prediction algorithm . . . 22

**D Complete results and fitting budgets** **23**
D.1 Individual surrogate, baseline, and ensemble errors . . . 23
D.2 Comparison with retrospectively selected models . . . 23
D.3 Fitting budgets and uniform averaging . . . 24
D.4 Selection and fitting within smaller model subsets . . . 24
D.5 Elo ranking . . . 25

**E Sensitivity analyses and ERA5 evaluation** **26**
E.1 Sensitivity to comparison scope . . . 26
E.2 Error reductions by PDE class . . . 26
E.3 Supplementary control for coarse pretraining at smaller fine data budgets . . . 27
E.4 ERA5 protocol and controls . . . 28
E.5 Additional fitting and evaluation cases for Heat I . . . 30

**F Ensemble error analysis and related methods** **31**
F.1 Error geometry and the fitting objective . . . 31
F.2 Error matrix estimation and excess risk . . . 32
F.3 Comparison with prior ensemble and multifidelity methods . . . 33

**G Reproducibility, AI use, and ethics statements** **34**
G.1 Reproducibility statement . . . 34
G.2 AI use statement . . . 34
G.3 Ethics statement . . . 34

# A Datasets and evaluation protocol

## A.1 Dataset definitions

Table 3 describes the 22 prediction tasks. Here $d$ counts input parameters, $N_H$ counts HF training rows, and $N_{L,\min}$ is the smallest LF table, not the sum over fidelities. Query inputs are excluded from every training fidelity and are subsequently partitioned into fitting and evaluation sets. These counts describe available data, rather than identical exposure under every training procedure. The seven PDE classes contain six elliptic, two diffusion, three convection, two shock, one porous flow, five reaction diffusion, and two wave datasets. ERA5 supplies the climate case and enters dataset aggregation, but not PDE class aggregation.

Table 3: **Dataset definitions and available training data.** Query counts include fitting pools and evaluation cases. Work grids are used for prediction and scoring. Citations identify imported data, while uncited entries use our simulation code.

| Dataset | $d$ | Levels | $N_{L,\min}$ | $N_H$ | Query | Work grid | Class |
|---|---|---|---|---|---|---|---|
| Darcy | 16 | 3 | 400 | 400 | 100 | $128 \times 128$ | Elliptic |
| Eikonal | 3 | 2 | 400 | 400 | 100 | $64 \times 64$ | Elliptic |
| Helmholtz | 2 | 3 | 400 | 400 | 100 | $256 \times 256$ | Elliptic |
| Poisson I | 5 | 3 | 400 | 400 | 100 | $64 \times 64$ | Elliptic |
| Poisson II (Niu et al., 2024) | 5 | 5 | 64 | 64 | 512 | $128 \times 128$ | Elliptic |
| Pressure Poisson | 2 | 4 | 400 | 400 | 100 | $64 \times 64$ | Elliptic |
| Heat I | 3 | 3 | 400 | 400 | 100 | $64 \times 64$ | Diffusion |
| Heat II (Niu et al., 2024) | 3 | 5 | 1024 | 1024 | 512 | $128 \times 128$ | Diffusion |
| Cavity | 1 | 4 | 400 | 400 | 100 | $256 \times 256$ | Convection |
| Navier Stokes (Niu et al., 2024) | 2 | 2 | 256 | 256 | 256 | $64 \times 64$ | Convection |
| Rayleigh Bénard | 1 | 2 | 400 | 400 | 100 | $64 \times 64$ | Convection |
| Burgers | 2 | 3 | 400 | 400 | 100 | $256 \times 256$ | Shocks |
| Euler | 7 | 3 | 400 | 400 | 100 | $128 \times 128$ | Shocks |
| Porous medium | 2 | 3 | 400 | 400 | 100 | $256 \times 256$ | Porous flow |
| Allen Cahn | 19 | 3 | 400 | 400 | 78 | $256 \times 256$ | Reaction diffusion |
| Cahn Hilliard I | 2 | 2 | 400 | 400 | 100 | $64 \times 64$ | Reaction diffusion |
| Cahn Hilliard II | 19 | 3 | 400 | 400 | 100 | $256 \times 256$ | Reaction diffusion |
| Fisher KPP | 50 | 3 | 400 | 400 | 100 | $256 \times 256$ | Reaction diffusion |
| Phase field crystal | 18 | 3 | 400 | 400 | 100 | $128 \times 128$ | Reaction diffusion |
| Shallow water | 2 | 3 | 400 | 400 | 100 | $128 \times 128$ | Waves |
| Wave | 3 | 2 | 400 | 400 | 100 | $80 \times 80$ | Waves |
| ERA5 (Hersbach et al., 2020, Niu et al., 2024) | 12 | 9 | 313 | 55 | 17 | $128 \times 256$ | Climate |

Poisson II, Heat II, and Navier Stokes use data released with MFRNP (Niu et al., 2024). ERA5 combines its climate model sources with reanalysis targets (Hersbach et al., 2020, Niu et al., 2024). Other entries were simulated for this collection. Pressure Poisson adapts an analytic construction motivated by Partin et al. (2022), Euler a quadrant setup motivated by The Well (Ohana et al., 2024), and Shallow water a radial dam break recipe motivated by PDEBench (Takamoto et al., 2022). These recipe influences do not make the independently simulated entries copies of those benchmarks. Euler, Burgers, and Shallow water call PyClaw (Ketcheson et al., 2012), using first order Godunov methods at coarse fidelities and SharpClaw WENO reconstruction at the fine fidelity. Our dam break varies both the inner water height and radius on a unit square, with independent solves at each grid. Phase field crystal implements the conserved dynamics of Elder & Grant (2004). Roman numerals distinguish tasks with the same equation. Poisson I and II have related parameterizations, while Cahn Hilliard I and II have 2 and 19 input parameters. Cahn Hilliard II, Allen Cahn, and Fisher KPP use $64 \times 64$, $128 \times 128$, and $256 \times 256$ fidelity grids. Phase field crystal uses $32 \times 32$, $64 \times 64$, and $128 \times 128$ grids. Their input dimensions are 19, 19, 50, and 18, respectively. Heat fields span space and time, and Helmholtz predicts amplitude normalized fields.

**Numerical checks and their scope.** Verification scripts test the custom implementations against analytical solutions, conservation laws, and refined numerical solutions. In a one dimensional Barenblatt test (Vázquez, 2007), the Porous medium solver's relative $L^2$ error falls from $8.1 \times 10^{-4}$ at 128 points to $2.6 \times 10^{-5}$ at 512 points. An Allen Cahn mesh check reduces error from $4.9 \times 10^{-5}$ at 64 points to $2.3 \times 10^{-6}$ at 256 points against a 512 point reference. The archived checks also test mass conservation for Cahn Hilliard II and Phase field crystal, and the Helmholtz discrete equation

residual. These are controlled checks of selected solver components, including one dimensional tests for solvers used in two dimensions. They do not certify every sampled parameter or the accuracy of every stored field. The companion package records the scripts, measured errors, and scope of each check. Published software and recipe references establish provenance, while the implementation checks are our own.

### A.2 Training, fitting, and evaluation cases

Fitting examples determine model choice or ensemble weights, while evaluation examples measure prediction error. Both sets are excluded from LF and HF training before preprocessing or model fitting. For each PDE dataset, we repeat the fitting and evaluation split three times without retraining the models. Within each split, all rules use the same examples. We test $K = 5$, 10, and 20 fitting examples, with each larger set containing the smaller ones. Within each split, the evaluation cases are fixed across all three budgets. One example pairs an input vector with its complete fine field. Model choices and weights are fixed before evaluation.

ERA5 uses 55 fine training cases, ten cases available for fitting, and seven fixed evaluation cases. At $K = 5$, we use three subsets of the ten fitting cases. At $K = 10$, we use all ten, so repeating the fit does not change the examples. Both budgets use the same seven evaluation cases. Table 3 provides training and reserved case counts for all datasets.

The repeated splits reuse trained models and some cases, so they measure sensitivity to the choice of fitting examples rather than independent training runs. Historical evaluation results informed development, making these comparisons retrospective. Fixed splits and scripts that reproduce the reported errors from saved predictions are included in the companion package.

### A.3 Field visualization conventions

Figure 1 uses the first archived ERA5 fitting split and its five example fitted mixture. Its error matrix uses only the fitting fields. The prediction is for the first evaluation input, selected independently of its error. The training fields share the same parameter vector and retain their native $192 \times 384$ and $721 \times 1440$ grids. Model predictions and their weighted sum retain the stored $128 \times 256$ working grid. Fields share a linear color scale, with no added blur or interpolation. The three stacked thumbnails represent nine predictions, all included in the weighted sum. The saved weights, source identities, and figure generation code accompany the figure.

Figure 2 shows row zero of each HF training table, chosen independently of prediction error. Its grids are native array grids, while the inventory gives the working grids used for scoring. Heat axes span space and time. The cavity target is vorticity, with the top lid moving right and array rows increasing upward. Its display uses a symmetric logarithmic scale, linear within $[-1, 1]$ and bounded by $[-400, 400]$, so wall values do not hide the interior structure. No values are clipped or modified. Enlarged panels, numerical display scales, extraction metadata, and source hashes for all 22 datasets are included in the companion package.

Figure 3 illustrates Heat I, selected after inspecting dataset results. It uses the first evaluation case of the first partition, not the case with the greatest improvement, and reuses the saved five example fits. All predicted fields share a linear color scale. Absolute errors share a logarithmic scale that is linear between zero and $10^{-5}$. Scores use the complete $64 \times 64$ fields without smoothing. The selected model is M4, so its column repeats that model.

# B MODEL ADAPTATIONS AND TRAINING

The library varies both the representation of a field and how coarse data enter learning. We first identify the retained published components and our adaptations, then give the training settings. Baseline descriptions specify the evaluated implementations and their settings.

## B.1 MODEL SOURCES, ADAPTATIONS, AND RATIONALE

Each library member combines a field representation with a training and inference procedure. We describe M1 to M9 individually below, identifying the published components retained, the changes made for parameter to field prediction, and the reason for including each model. Several members combine ideas from more than one paper. The resulting library is an integration of adapted models, not nine independent reproductions or nine newly proposed architectures. All accept parameters alone at deployment. Appendix B.2 gives their training budgets and data use. The rationales below motivate the library's composition, while the experiments assess its predictions. They do not isolate the effect of every architectural change.

**M1: FiLM FNO transfer.** The starting components are FNO's learned Fourier mixing (Li et al., 2021), FiLM's featurewise affine conditioning (Perez et al., 2018), and the LF pretraining followed by HF finetuning strategy of Lyu et al. (2023). We retain spectral convolutions with local residual paths and adapt the input interface to a simulation parameter vector. The spatial input contains coordinates only. At each block, a parameter network generates a scale and shift applied after group normalization, so the parameters influence features throughout the network. Related parameter conditioning for PDE emulation is studied by Shokar et al. (2025). Our model differs from our parameter concatenation FNO references, which supply parameters as constant spatial channels at the input. The same weights are first trained on LF fields and then finetuned on HF fields. This combination tests whether repeated parameter conditioning and a coarse initialization provide an effective way to generate globally coupled fields.

**M2: All pairs FNO.** M2 retains M1's FNO and FiLM components (Li et al., 2021, Perez et al., 2018) and adapts the idea of pair enumeration from Poseidon (Herde et al., 2024). Poseidon uses pairs of states for temporal operator learning. Here the pairs index source and target fidelities. We append their normalized labels to the parameter vector, pool the corresponding target fields on a common working grid, and train one conditional FNO before HF finetuning. The source label specifies a training pair, but no source field enters the network. Poseidon's architecture, pretrained weights, and temporal operator are not used. For ERA5, exact input joins retain each target field with its own parameters. The purpose is to let one model share information across fidelity tasks before specializing to the fine field. Pair enumeration also repeats targets and changes optimization cost, so its effect cannot be attributed to fidelity conditioning alone.

**M3: ConvNeXt transfer.** ConvNeXt supplies the depthwise spatial convolution, channel expansion, and residual block structure (Liu et al., 2022). We place these blocks in an U shaped encoder and decoder with skip connections (Ronneberger et al., 2015) and a dense field output, adapting the image classification backbone to field generation. The model starts from coordinates and replaces the usual block normalization with group normalization and FiLM conditioning on the simulation parameters (Perez et al., 2018). Three downsampling stages and a bottleneck process the field at several resolutions, with corresponding decoder stages restoring the target grid. LF pretraining and HF finetuning follow the transfer principle of Lyu et al. (2023). The rationale is to provide local spatial processing alongside M1's Fourier mixing, while retaining the same type of parameter conditioning and transfer schedule.

**M4: Distribution FNO.** FIRE conditions a fine correction on summaries of a coarse predictive distribution using tabular foundation models and in context inference (Yu et al., 2026). M4 retains the distribution conditioning idea and replaces those models with FNOs trained on spatial fields (Li et al., 2021). Five LF members predict a coarse field at the query parameters. Their pointwise mean, standard deviation, and 10th, 50th, and 90th percentiles become spatial input channels for a residual FNO, together with coordinates and broadcast parameters. The residual is added to the coarse mean to predict the fine field. Both stages are trained by gradient optimization, so this implementation

does not retain FIRE's inference without model retraining. The ensemble summaries are features of prediction variability, not a guarantee of accurate uncertainty estimates. The rationale is to expose coarse disagreement as well as the mean to the correction model. This recipe has no out of fold construction and is distinct from the coarse ensembles used by M8 and M9.

**M5: Wavelet transfer.** WNO motivates learning in wavelet coordinates (Tripura & Chakraborty, 2022). We replace M1's Fourier mixing with a custom single level db4 wavelet transform, while retaining its coordinate input, FiLM conditioning (Perez et al., 2018), local residual paths, and LF to HF transfer schedule (Lyu et al., 2023). Within each wavelet subband, learned weights mix channels at coefficient positions inside a capped leading block. Other positions pass through unchanged before the inverse transform. These indices describe positions within a subband, not a Fourier frequency cutoff. The implementation also handles rectangular grids, odd grid sizes, and one dimensional grids. It is a custom wavelet representation rather than an exact reproduction of the published WNO. Its inclusion tests whether spatially localized wavelet features provide useful predictions alongside the spectral and convolutional transfer models.

**M6: Retrieved field DeepONet.** DeepONet supplies the branch and trunk representation (Lu et al., 2021), while the multifidelity use of a coarse prediction as a reference is motivated by Lu et al. (2022). We use a Cartesian product DeepONet whose branch receives the parameters concatenated with a stored LF training field and whose trunk receives target grid coordinates. A nearest neighbour search in standardized parameter space selects that field. The network learns a residual relative to its interpolated version, which is then added back to form the fine prediction. Retrieval is the deployment adaptation: a new query requires parameters only and no new coarse solve. The rationale is to exploit a nearby physical field without first learning a separate coarse surrogate. If an exact LF match exists at a training input, retrieval supplies that match, whereas an unseen query uses a neighboring input. This difference in reference quality remains a limitation. The implemented residual model is also distinct from the composite multifidelity architecture of Howard et al. (2023).

**M7: POD GP.** Basis emulation represents a large field through a small set of coefficients (Higdon et al., 2008). M7 retains this principle, constructing a POD basis by singular value decomposition of centered HF training fields. A Gaussian process maps parameters to each retained coefficient, and the predicted coefficients are projected back through the basis and added to the training mean field. Our implementation uses an RBF kernel with automatic relevance determination and shares covariance hyperparameters across coefficient outputs. It does not implement the full Bayesian hierarchy of the cited model. M7 is deliberately fine only: no coarse field enters basis construction, fitting, or inference. It supplies a compact alternative when the field varies mainly in a few directions or when learning a coarse correction is less effective than learning the fine coefficients directly.

**M8: Slice attention corrector.** Transolver motivates grouping spatial cells into learned slice tokens (Wu et al., 2024). We retain pooled spatial summaries but change the attention operation. Original Transolver normalizes each cell's assignments across slices, applies attention between slice tokens, and maps back using the assignments. Our pooling weights instead normalize across cells, and cell queries attend directly to the pooled slice tokens. The model receives the current field, coordinates with sinusoidal features, and FiLM conditioning on the parameters (Perez et al., 2018). A coarse surrogate ensemble supplies its initial state, with out of fold predictions for training cases. A head initialized to zero predicts additive corrections through six damped updates with shared weights. Training penalizes spatial squared error and Fourier magnitude discrepancy at each update, plus a nonzero update when started at the target field. This encourages an equilibrium without guaranteeing contraction. The adaptation tests whether pooled global interactions can refine a predicted coarse field using parameters alone at deployment. Its attention and iterative recipe differ from the published Transolver.

**M9: ConvNeXt corrector.** M9 retains ConvNeXt's spatial and channel mixing blocks (Liu et al., 2022) and adapts them to the same correction task as M8. It uses a custom U shaped encoder and decoder with skip connections (Ronneberger et al., 2015), three downsampling stages and a bottleneck, and FiLM parameter conditioning (Perez et al., 2018). Unlike M3, which generates a field from coordinates and parameters after transfer training, M9 receives the current coarse estimate

as a spatial input and outputs an additive update. Its head is initialized to zero, making the initial refinement the identity. The coarse ensemble construction, six damped updates, shared weights, and spatial, spectral, and target state penalties follow the M8 recipe. The rationale is to offer local processing at several resolutions as an alternative to pooled attention when correcting the same kind of coarse prediction. This is a ConvNeXt based iterative corrector, not the original classification model.

For comparison, we use baseline implementations based on the cited methods, with parameter inputs and field outputs. The descriptions below specify the models and training procedures used to produce the reported predictions.

**B1, B2, and B11: classical multifidelity field references.** B1 uses autoregressive discrepancy modeling (Kennedy & O'Hagan, 2000) with POD GP field representations. One scalar coefficient couples the LF prediction to HF data, and a separate POD GP predicts the residual. Inference combines their posterior means. B2 models nonlinear dependence on LF coefficients (Perdikaris et al., 2017). Its HF kernel combines input dependent coupling, an LF coefficient kernel, and a discrepancy kernel, with covariance hyperparameters shared across HF outputs. Monte Carlo samples propagate marginal LF uncertainty. B11 uses an affine field correction (Zhang et al., 2018): one slope and intercept are fitted over HF training cases and pixels, then applied to the LF POD GP prediction. Paired training cases use observed LF fields, while deployment uses predicted LF fields.

**B3, B4, and B5: neural fusion references.** B3 uses composite linear and nonlinear correlation modeling (Meng & Karniadakis, 2020). A tanh MLP first maps parameters and coordinates to LF values. After this stage is frozen, linear and nonlinear HF networks learn from the inputs and LF prediction, with weight decay on the nonlinear branch. B4 uses composite DeepONets (Howard et al., 2023): a frozen LF DeepONet predicts coarse sensors, which enter linear and nonlinear HF branches alongside the parameters. B5 is a deterministic hierarchical latent network based on the transfer idea of Li et al. (2022a). Two encoders and linear field decoders are trained jointly with LF and HF squared losses. The second encoder receives both the parameters and the first latent vector.

**B6 and B7: alternative decoder and probabilistic models.** B6 combines NOMAD's nonlinear decoding principle (Seidman et al., 2022) with LF pretraining and HF finetuning. A parameter MLP produces a latent vector, and a coordinate conditioned MLP decodes the field. B7 uses MFRNP's upstream `MultiFidelityModel` (Niu et al., 2024), including its probabilistic residual model, likelihood loss, and latent regularization. Its data adapter supports rectangular grids with a maximum side of 256 and bilinear antialiased resizing. A scalar mean and standard deviation from the first fidelity normalize inputs, while outputs are scaled per fidelity. Fidelities are shuffled independently, and the smallest table limits batch size. Hidden width is 128, latent width is 64, and the model has three hidden layers. Adam uses learning rate and epsilon $10^{-3}$. HF and LF likelihood weights are two and one. Context fractions are $[0.20, 0.25]$, increasing to $[0.40, 0.60]$ when HF training has at most ten examples. In that case all training rows are retained. Prediction uses the final checkpoint and latent summaries from the final batch.

**B8, B9, and B10: fidelity and transfer FNO references.** B8 uses the fidelity dependent representation principle of IFC (Li et al., 2022b) with an FNO spatial basis. Weights predicted from the fidelity features $[\ell, \ell^2]$ mix its latent channels. Training uses maximum absolute scaling per fidelity, LF warm up, and joint LF and HF finetuning. Intermediate fidelities use the nearest registered scaler. B9 and B10 use parameter concatenation FNO transfer (Lyu et al., 2023). They are two result entries of the same architecture and training procedure. Historical entries contain separate fits. On ERA5 and the four additional reaction diffusion tasks, both entries use the same fitted predictions.

### B.2 Training settings and data use

Table 4 summarizes the library schedules. For M1, M2, M3, and M5, AdamW uses batch size at most 16, weight decay $10^{-5}$, gradient clipping at 1, and cosine decay to $10^{-6}$. Initial learning rates are $10^{-3}$ for pretraining and $3 \times 10^{-4}$ for HF finetuning. The FNO recipe uses width 64, four blocks, and up to 12 Fourier modes per direction, subject to the grid cap. M4 also uses AdamW but starts both its LF members and residual stage at $10^{-3}$. M6 uses width 256 and five affine layers of that

width in each branch and trunk specification, including the latent output, with ReLU and Adam at $10^{-3}$. Detailed grid dependent definitions and fitted hyperparameters are retained with the model implementations and metadata.

Table 4: **Library training schedules.** Epochs count passes through the relevant training table. Optimizer steps and repeated refinement steps are different quantities. Each final surrogate uses a single seed, with internal coarse ensemble initializations prescribed by its recipe.

| Models | Nominal training procedure |
|---|---|
| M1, M3, M5 | 2,500 LF epochs, followed by 2,500 HF epochs. |
| M2 | 2,500 epochs over pooled fidelity pairs, followed by 1,250 HF epochs. |
| M4 | Five LF FNOs and one HF residual FNO, each trained for 2,500 epochs. |
| M6 | 125,000 Adam iterations on HF residual targets. |
| M7 | HF basis construction and coefficient GP fitting, without an epoch schedule. |
| M8, M9 | Coarse members request 30,000 optimizer steps, rounded to whole epochs. Each corrector uses 6,000 optimizer steps and six refinement updates per prediction. |

M1, M3, M4, M5, and M6 use the lowest registered LF level. M2 pools pairs across registered fidelities, M8 and M9 predict the finest registered LF level, and M7 uses HF fields only. ERA5 consequently uses level one for the transfer and retrieval recipes and level eight for the iterative correctors. For M8 and M9, five LF folds each contain four members: plain and heteroscedastic variants with two initializations each. Training cases receive the mean from their held out fold. All query cases use the four members of fixed fold zero. Corrector checkpoints are selected using training only validation. M7's basis uses only HF base training fields. Its original estimator state was not saved, so the common prediction export reconstructs that training only fit and restores target scaling.

Baselines use the settings of the evaluated benchmark implementations, with the adaptations above. Training, validation allocations, and computation vary across recipes. For example, on Darcy, B3 to B5 use 360 fine rows for fitting and 40 for internal validation, while B1, B2, and the transfer recipe fit 400 rows. Both allocations consume 400 fine labels before ensemble fitting. The total fine data budget is $N_H^{\mathrm{base}} + K$, plus any additional HF labels used for tuning, where $N_H^{\mathrm{base}}$ counts base training labels and $K$ counts ensemble fitting examples. Final evaluation labels are separate. The library reuses the available simulation data across candidates. Each ensemble still requires the trained candidates and their coarse members, even when a final weight is zero. The comparisons use common evaluation cases under the stated recipes. Training resources are not matched across methods.

# C ENSEMBLE IMPLEMENTATION

## C.1 NUMERICAL OPTIMIZATION AND COMPUTATIONAL COST

The Selected model, Inverse error mixture, and Fitted mixture use the same $K$ input and fine field pairs. Their definitions appear in Section 3. For joint fitting, the empirical error matrix is symmetrized and divided by $\max(\mathrm{tr}(\widehat{C})/M, 10^{-30})$. Here the trace sums diagonal errors and $M = 9$ counts models. This positive scaling changes the objective magnitude without changing the matrix condition number or the exact minimizer. SLSQP uses the analytic gradient, bounds $0 \leq w_m \leq 1$, the unit sum constraint, tolerance $10^{-12}$, and at most 300 iterations. Uniform weights and the lowest diagonal singleton initialize separate solves. Finite solutions with unit sum residual below $10^{-6}$ are clipped to nonnegative values and renormalized. Feasible starting points remain candidates, and the lowest fitting objective determines the retained weights. These checks establish numerical feasibility, not generalization to new inputs.

Uniform averaging, used as a control in Table 7, assigns every member weight $1/M$. Forming the error matrices costs $O(KM^2P)$ for $P$ values per field. Storing the fitting matrices requires $O(KM^2)$ memory, excluding prediction buffers. These costs exclude surrogate training and inference. The companion package preserves additional regularization, pair fitting, and coarse input refresh experiments separately from the three main rules.

## C.2 AUTOMF TRAINING AND PREDICTION ALGORITHM

**Algorithm 1: train models, fit the ensemble, and predict**
**Input:** LF/HF training tables, $K$ reserved fine cases, fixed model configurations, and one prescribed rule: Selected model, Inverse error mixture, or Fitted mixture.
1. Check input identities and working grids within the dataset. Train the surrogate models using training data only.
2. Predict fields for the fitting inputs and compute $G_1, \ldots, G_K$.
3. Fit the prescribed rule on all $K$ cases using the definitions in Section 3.
4. Save preprocessing, model identities, and fitted weights.
**Predict:** combine model predictions at new parameters using Equation (1).
**Evaluate:** score a separate evaluation set only after saving these choices.

The algorithm separates surrogate training from ensemble fitting. The bundled program implements steps 2 to 4 and combines saved model predictions for new inputs. Dataset adapters and base model training remain separate components. The three rules use the same fitting and evaluation cases within each comparison.

# D COMPLETE RESULTS AND FITTING BUDGETS

## D.1 INDIVIDUAL SURROGATE, BASELINE, AND ENSEMBLE ERRORS

Table 1 reports all nine library members and the three ensemble rules on every dataset. Table 5 completes that comparison with individual baseline errors. The latter include the parameter kNN and training mean controls, which are outside the ranked B1 to B11 collection. Every ranked method covers all 22 datasets. Errors average the same evaluation cases within each partition, followed by the three partitions. ERA5 uses its separate seven case evaluation set. Full precision values, per case errors, and replay metadata are retained in the companion package.

Table 5: **Baseline errors across all 22 datasets.** Mean relative $L^2$ errors in percent. B1: Autoregressive POD GP. B2: Nonlinear POD GP. B3: Composite MF MLP. B4: Composite MF DeepONet. B5: Latent MF network. B6: Nonlinear decoder transfer. B7: MFRNP adapter. B8: Fidelity basis FNO. B9 and B10: FNO transfer I and II. B11: Affine POD GP. B12: Parameter kNN. B13: Training mean. Mixtures use five fitting examples and the same evaluation cases. A dash marks an unavailable additional control, not a missing ranked method. B9 and B10's implementation relationship is described in Appendix B.1. Imported data are from MFRNP (Niu et al., 2024) and ERA5 reanalysis (Hersbach et al., 2020).

| Dataset | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | B12 | B13 | Inverse error mixture | Fitted mixture |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Darcy | 3.37 | 3.66 | 4.03 | 6.66 | 5.91 | 5.2 | 12.4 | 5.55 | 5.27 | 5.28 | 3.6 | 21.5 | 38 | 2.55 | 2.33 |
| Eikonal | 3.49 | 2.99 | 3.12 | 3.35 | 3.38 | 2.65 | 3.62 | 2.21 | 2.33 | 2.34 | 4.88 | 5.51 | 31.8 | 1.76 | 1.76 |
| Helmholtz | 12.3 | 15.2 | 11.5 | 7.84 | 8.12 | 99.6 | 58.6 | 99.7 | 4.12 | 4.12 | 15.9 | 25.7 | 92 | 6.89 | 6.55 |
| Poisson I | 0.00731 | 0.00531 | 0.181 | 0.161 | 0.1 | 0.479 | 6.79 | 0.239 | 0.0804 | 0.0806 | 2.68 | 7.51 | 31.9 | 0.008 | 0.00788 |
| Poisson II | 0.0128 | 0.0306 | 0.635 | 1.72 | 0.849 | 3.93 | 11.7 | 82.1 | 0.355 | 0.357 | 2.01 | 12.3 | 38.5 | 0.0125 | 0.0129 |
| Pressure Poisson | 0.457 | 0.472 | 0.07 | 1.53 | 0.127 | 0.446 | 1.05 | 0.684 | 0.0639 | 0.0645 | 0.681 | 0.196 | 1.58 | 0.0769 | 0.0802 |
| Heat I | 0.326 | 0.327 | 0.0696 | 0.125 | 0.069 | 0.432 | 1.29 | 0.275 | 0.0207 | 0.0206 | 1.09 | 1.35 | 12.4 | 0.00593 | 0.0062 |
| Heat II | 0.312 | 0.312 | 0.111 | 0.183 | 0.0721 | 0.219 | 7.4 | 0.32 | 0.0202 | 0.0201 | 0.405 | 0.856 | 13.2 | 0.00437 | 0.00435 |
| Cavity | 0.307 | 0.409 | 23.9 | 4.99 | 0.0419 | 24.1 | 55.7 | 0.129 | 0.622 | 0.666 | 37.2 | 0.077 | 17.1 | 0.142 | 0.142 |
| Navier Stokes | 2.8 | 2.48 | 2.08 | 4.01 | 3.71 | 7.08 | 25.7 | 1.62 | 1.52 | 1.52 | 21.6 | 3.27 | 41.2 | 1.52 | 1.5 |
| Rayleigh Bénard | 0.283 | 0.295 | 0.0542 | 0.0588 | 0.0134 | 0.0613 | 4.5 | 0.78 | 0.0113 | 0.011 | 0.715 | 0.0575 | 9.1 | 0.00286 | 0.00269 |
| Burgers | 2.16 | 2.5 | 1.54 | 8.14 | 1.22 | 15.7 | 27.7 | 0.928 | 0.562 | 0.562 | 11.7 | 0.677 | 73.2 | 0.513 | 0.502 |
| Euler | 5.6 | 5.54 | 6.56 | 6.16 | 6.15 | 8.81 | 9.15 | 7 | 6.52 | 6.43 | 6.25 | 6.21 | 32.2 | 5.64 | 5.73 |
| Porous medium | 1.15 | 0.859 | 0.299 | 1.31 | 0.772 | 0.918 | 33.7 | 5.14 | 0.0718 | 0.0718 | 1.98 | 1.92 | 31.9 | 0.0262 | 0.0263 |
| Allen Cahn | 55.5 | 98.6 | 59.9 | 44.8 | 41.2 | 94.4 | 69 | 48.9 | 55.8 | 55.8 | 55.4 | 78.3 | 100 | 37.9 | 34.6 |
| Cahn Hilliard I | 29.7 | 27.8 | 53.5 | 48.7 | 27.6 | 61.1 | 74.7 | 14.2 | 13.4 | 13.3 | 72.3 | 20.6 | 99.9 | 11.3 | 10.7 |
| Cahn Hilliard II | 64.8 | 83.5 | 85 | 53.7 | 49.8 | 72.4 | 66.7 | 54.4 | 54 | 54 | 64.8 | 78 | 100 | 47.2 | 49.3 |
| Fisher KPP | 6.86 | 6.86 | 6.48 | 4.01 | 3.45 | 7.44 | 4.08 | 4.74 | 3.24 | 3.24 | 6.86 | 5.67 | 6.86 | 2.53 | 2.3 |
| Phase field crystal | 83.5 | 86.4 | 90.7 | 87.5 | 87.8 | 152 | 89.4 | 102 | 86.8 | 86.8 | 83.5 | 86.7 | 87.5 | 83.8 | 83.3 |
| Shallow water | 1.14 | 1.02 | 0.467 | 0.878 | 0.691 | 0.883 | 5.59 | 0.806 | 0.403 | 0.403 | 2.94 | 1.22 | 25.1 | 0.357 | 0.389 |
| Wave | 27.7 | 24.9 | 3.3 | 42.3 | 46.6 | 5.39 | 93.9 | 25.9 | 6.6 | 6.6 | 50.3 | 42.8 | 98.7 | 4.39 | 4.27 |
| ERA5 | 5 | 6.74 | 45 | 16.8 | 5.05 | 23.9 | 19.5 | 5.15 | 11.9 | 11.9 | 8.45 | 5.16 | 6.74 | 5.81 | 6.49 |

## D.2 COMPARISON WITH RETROSPECTIVELY SELECTED MODELS

Two comparisons separate the quality of the available library from the benefit of combining it. The best library reference minimizes partition averaged error over M1 to M9 separately for each dataset. The best baseline minimizes the corresponding error over B1 to B11. Both use evaluation answers retrospectively and are stronger references than the deployable Selected model, which uses only fitting examples. Table 6 reports these comparisons on the 21 PDE datasets. Table 2 additionally includes ERA5 in its dataset ratios and Elo ranking.

A still stronger control chooses a potentially different best model in each partition using its evaluation answers. Against this reference, the fitted mixture at $K = 5$ has class ratio 0.933 and dataset ratio 1.006, with lower error on 13 tasks and higher error on eight. Thus its advantage over retrospective model choice depends on both the reference and aggregation.

Table 6: **Library quality and mixture gains on the 21 PDE datasets.** Ratios are geometric means. Wins and losses count datasets with lower and higher error, respectively. Best library and best baseline refer to the retrospective per dataset minima described above.

| Comparison | Class ratio | Dataset ratio | Wins | Losses |
|---|---|---|---|---|
| Best library / best baseline | 0.766 | 0.834 | 16 | 5 |
| Fitted mixture / best library | 0.923 | 0.989 | 14 | 7 |
| Inverse error mixture / best library | 0.927 | 1.002 | 12 | 9 |
| Fitted mixture / best baseline | 0.706 | 0.825 | 14 | 7 |
| Fitted mixture / Selected model | 0.884 | 0.923 | 16 | 5 |

### D.3 Fitting budgets and uniform averaging

Table 7 compares the main rules with uniform averaging on the 21 PDE datasets. Table 8 varies the fitting budget while keeping the models and case partition protocol fixed. Each ratio compares methods that receive the same number of fitting examples. These summaries measure sensitivity to ensemble fitting, not independent training seed variation.

Table 7: **Weighting the same nine model library.** Ratios compare with the Selected model at $K = 5$ on 21 PDE datasets. Lower is better. Wins and losses count datasets with lower and higher error, respectively. Exact ties count as neither.

| Method | Class balanced | Equal dataset | Wins | Losses |
|---|---|---|---|---|
| Selected model | 1.000 | 1.000 | 0 | 0 |
| All pairs FNO | 1.333 | 1.502 | 4 | 15 |
| Uniform (all nine) | 5.919 | 4.944 | 2 | 19 |
| Inverse error mixture | 0.888 | 0.935 | 14 | 7 |
| Fitted mixture | 0.884 | 0.923 | 16 | 5 |

Table 8: **Increasing the fitting budget on the 21 PDE datasets.** Class aggregated error ratios. The models remain fixed, and every comparison uses the same allowance of $K$ additional fine fields.

| $K$ | Inverse error mixture / Selected model | Fitted mixture / Selected model | Fitted mixture / Inverse error mixture |
|---|---|---|---|
| 5 | 0.8876 | 0.8840 | 0.9958 |
| 10 | 0.8921 | 0.8765 | 0.9826 |
| 20 | 0.8900 | 0.8692 | 0.9766 |

### D.4 Selection and fitting within smaller model subsets

These existing controls use the same frozen predictions, five fitting examples, and evaluation cases as the main rules on the 17 PDE diagnostic tasks. They exclude the four subsequently added reaction diffusion tasks and ERA5. The best two and best three subsets rank models by mean relative $L^2$ error on the fitting cases. Their uniform variants assign equal weights within the subset. The fitted best three variant instead minimizes Equation (4) over those three models. The fitted pair searches all model pairs and their convex weights using the squared fitting error, retaining a single model if no pair improves that objective. Subset choices and weights use no evaluation answers. Every rule is scored on the same cases after fitting.

Removing poorly performing candidates makes uniform averaging much more competitive than averaging all nine. However, the best three uniform rule has class ratio 0.997 and dataset ratio 1.178, compared with 0.876 and 0.918 for the full fitted mixture. Fitting the best three gives 0.889 and 0.923, recovering much of the benefit. The candidate pool therefore provides useful choices without establishing that every task needs all nine predictions. These controls retain a heterogeneous trained library. They do not compare it with repeated training of one architecture or an ensemble of baseline implementations, and selecting subsets still requires their fitting predictions.

Table 9: **Model subset controls on 17 PDE tasks.** Error ratios compare with the Selected model using the same five fitting examples. Both columns aggregate per dataset ratios geometrically, either giving PDE classes equal weight or giving datasets equal weight. Lower is better. Best refers to fitting error, not evaluation error.

| Rule | PDE class ratio | Dataset ratio |
|---|---|---|
| Uniform over all nine | 6.190 | 7.095 |
| Uniform over best two | 1.028 | 1.186 |
| Uniform over best three | 0.997 | 1.178 |
| Fitted pair | 0.926 | 0.954 |
| Fitted best three | 0.889 | 0.923 |
| Inverse error mixture | 0.882 | 0.924 |
| Fitted mixture | 0.876 | 0.918 |

### D.5 ELO RANKING

Each pair of methods plays one comparison per dataset using its unrounded partition averaged error. Lower error wins and an absolute difference below $10^{-12}$ is a tie. The nine library members, eleven baseline entries, and three rules all enter on the same 22 datasets. Ratings start at 1500. For ratings $R_a, R_b$, the expected score and update are

$$p_a = \frac{1}{1 + 10^{(R_b - R_a)/400}}, \qquad R'_a = R_a + 32(s_a - p_a), \qquad R'_b = R_b - 32(s_a - p_a). \tag{7}$$

Here $s_a$ is one for a win, zero for a loss, and one half for a tie. Primes denote updated ratings. We reset ratings and independently shuffle comparisons 500 times, then report the mean final rating. The code fixes the random orders. Sorting uses unrounded ratings and display uses integers. Datasets receive equal weight. Elo summarizes how often a method wins rather than the magnitude of improvement, statistical significance, or training cost.

# E SENSITIVITY ANALYSES AND ERA5 EVALUATION

## E.1 SENSITIVITY TO COMPARISON SCOPE

These diagnostics cover 17 PDE tasks. They were not extended to Allen Cahn, Cahn Hilliard II, Fisher KPP, Phase field crystal, or the ERA5 climate task. All 22 datasets remain in the main results. The diagnostics retain the trained models, predictions, and fitting cases. The 14 task data quality subset excludes Poisson II, Cahn Hilliard I, and Pressure Poisson. A second subset removes Heat II and Burgers, the two tasks whose B8 case correspondence was established through checkpoint evaluation. Their intersection contains 12 tasks. Class aggregation normalizes over represented classes. These retrospective sensitivity checks do not certify solver validity or provide independent replication.

Table 10: **Sensitivity to dataset inclusion.** Each cell gives the PDE class ratio followed by the dataset ratio. $N$ counts tasks. Library and baseline refer to retrospective best models after averaging partitions. Both mixtures use five fitting examples.

| Scope | $N$ | Library / baseline | Fitted mixture / Selected model | Inverse error mixture / Selected model |
|---|---|---|---|---|
| Diagnostic set | 17 | 0.764 / 0.820 | 0.876 / 0.918 | 0.882 / 0.924 |
| Data quality subset | 14 | 0.747 / 0.810 | 0.873 / 0.900 | 0.875 / 0.908 |
| Without Heat II and Burgers | 15 | 0.790 / 0.865 | 0.872 / 0.935 | 0.872 / 0.940 |
| Intersection | 12 | 0.776 / 0.865 | 0.869 / 0.918 | 0.865 / 0.926 |

Gains over the Selected model persist across these scopes. Superiority to the best individual library member is less stable. On the 15 task subset, the fitted mixture has class ratio 0.916 and dataset ratio 1.020 against that reference, improving under one aggregation and worsening under the other. The complete subset and budget tables remain in the companion package.

On Darcy, internal validation retained the initial zero correction state for both M8 and M9, so both return the same prolonged, scaled coarse prediction. Their archived predictions are identical and add no distinct prediction diversity on this dataset. Redistributing weight between these entries leaves the combined field unchanged, but inverse error weighting counts the shared prediction twice before normalization. Effective model count therefore measures weight concentration over entries, not independent mechanisms. We retain the evaluated library. These weights do not uniquely measure model usefulness.

## E.2 ERROR REDUCTIONS BY PDE CLASS

Table 11 decomposes the primary comparison using the same seven classes and unrounded errors as Table 2. Within each class, datasets receive equal weight in a geometric mean. The best baseline is selected retrospectively from B1 to B11 for each dataset. The fitted mixture has lower aggregate error than selection in six classes and than the best baseline in five. The largest baseline improvements are in Diffusion and Porous flow, while Elliptic and Waves favor the baseline reference. Thus the direction of improvement extends beyond the classes that contribute most to its aggregate magnitude.

Without the Diffusion and Porous flow tasks, the class error reduction is 4.3% against selection and 1.3% against the best implemented baseline, compared with 11.6% and 29.4% on the full PDE roster. These diagnostic omissions illustrate concentration of gain rather than define a replacement benchmark. They reuse the reported results and provide no independent evaluation. ERA5 enters the separate equal dataset aggregate, not this decomposition of PDE classes.

Table 11: **Class contributions to the fitted mixture's error ratios.** All rules use five fitting examples. Lower is better. All seven classes reproduces the primary PDE class aggregation. Other five classes is a sensitivity calculation omitting Heat I, Heat II, and Porous medium and renormalizing over the remaining classes. It does not change the main roster.

| Class | Tasks | Fitted / Selected | Fitted / Best baseline |
|---|---|---|---|
| Elliptic | 6 | 1.041 | 1.087 |
| Diffusion | 2 | 0.648 | 0.255 |
| Convection | 3 | 0.895 | 0.932 |
| Shocks | 2 | 0.956 | 0.961 |
| Porous flow | 1 | 0.810 | 0.367 |
| Reaction diffusion | 5 | 0.932 | 0.860 |
| Waves | 2 | 0.967 | 1.118 |
| All seven classes | 21 | 0.884 | 0.706 |
| Other five classes | 18 | 0.957 | 0.987 |

### E.3 Supplementary control for coarse pretraining at smaller fine data budgets

An existing training budget sweep compares the same FiLM FNO architecture trained on fine data alone (R0) with low fidelity pretraining followed by fine tuning (R1). We report 11 historical task variants corresponding to Poisson I, Heat I, Eikonal, Rayleigh Bénard, Wave, Cavity, Allen Cahn, Cahn Hilliard II, Helmholtz, Phase field crystal, and Porous medium. Only tasks in the current roster with a compatible coarse training pool are included. R1 uses 4,000 coarse training examples for ten tasks and 400 for Cavity, as recorded in the run metadata. These historical variants and training allocations are separate from the main comparison and do not establish a matched control for its case identities.

The fine training budget is $N_H \in \{20, 50, 100, 200, 400\}$. Both arms receive the same fine subset and evaluation set within each comparison. Fine subsets are nested within each of two seeds, and the budget construction preserves the source evaluation set. The seeds change both model initialization and subset sampling. R0 uses approximately 30,000 fine training updates, whereas R1 adds approximately 30,000 coarse training updates before the same fine update allowance. Their fine training learning rates are $10^{-3}$ and $3 \times 10^{-4}$, respectively. Thus this control compares two training procedures with the same fine data allowance, without isolating coarse data from training duration and schedule.

Table 12: **Individual FiLM FNO transfer across fine training budgets.** $N_H$ counts fine training examples. Each comparison is one historical task and seed with both R0 and R1 results. The error ratio is the geometric mean of R1's mean relative $L^2$ error divided by R0's, weighting available comparisons equally. Lower is better. One R1 result is unavailable for Phase field crystal at $N_H = 200$. The final row pools all available comparisons, which reuse tasks and evaluation cases across budgets.

| $N_H$ | Comparisons | R1 lower error | Error ratio R1 / R0 |
|---|---|---|---|
| 20 | 22 | 22 | 0.2787 |
| 50 | 22 | 22 | 0.3079 |
| 100 | 22 | 20 | 0.4992 |
| 200 | 21 | 19 | 0.7099 |
| 400 | 22 | 21 | 0.7623 |
| All | 109 | 104 | 0.4693 |

R1 has lower error in 104 of 109 available comparisons, with the largest aggregate gains at the smaller fine training budgets (Table 12). This supports coarse pretraining as a useful option for an individual surrogate in these settings. It does not compare the nine model ensemble with a matched fine only ensemble or establish an advantage at equal total computation. No method in this supplement receives a true coarse solution at the query input.

### E.4 ERA5 PROTOCOL AND CONTROLS

**Climate task and source data.** The released climate benchmark of Niu et al. (2024) maps emissions drivers to global two metre air temperature. $CO_2$ and $CH_4$ contribute one feature each, and black carbon and $SO_2$ contribute five principal component scores each, following the feature construction of ClimateBench (Watson-Parris et al., 2022). Thirteen ScenarioMIP simulators supply the coarse fields, including CESM2, UKESM1, and NorESM2. They are grouped into eight LF resolutions, with ERA5 as the ninth fidelity (Hersbach et al., 2020, Niu et al., 2024). These fidelities therefore combine different simulators as well as spatial resolutions. The finest LF grid is $192 \times 384$ and the released ERA5 grid is $721 \times 1440$. The published ERA5 protocol uses 1950 to 2014 for training and 2015 to 2021 for evaluation (Niu et al., 2024, Section 5.2).

**Case identities and prediction alignment.** The released HF tables contain 65 training rows and seven test rows. We train on the first 55 training rows, reserve the remaining ten for ensemble fitting, and retain the seven original test rows. All nine prediction archives have the same ordered query inputs. The evaluation targets exactly reproduce the released test fields after the recorded bilinear resize to $128 \times 256$. Input identity checks verify that no fitting or evaluation input remains in training at any fidelity. These checks establish correspondence within the released arrays, without assigning calendar labels to individual rows.

M8 and M9 use 49 fine training cases for fitting and six for internal validation, with four coarse predictions per case assembled by the procedure in Appendix B.2. The twelve completed baseline entries represent eleven separately trained models because B10 reuses B9, and the training mean provides a separate control. Ranked baselines B1 to B11 all have scores on the same seven evaluation cases.

Table 13: **ERA5 mixtures and controls at two fitting budgets.** Mean relative $L^2$ errors in percent. At $K = 5$, three fitting selections use subsets of the same ten case pool. At $K = 10$, they repeat the complete pool. Both budgets reuse the same seven evaluation cases and trained models.

| Method | $K = 5$ | $K = 10$ |
|---|---|---|
| Inverse error mixture | 5.8146 | 5.8358 |
| Fitted mixture | 6.4877 | 6.4736 |
| Selected model (POD GP) | 6.7369 | 6.7369 |
| Training mean field | 6.7369 | 6.7369 |
| Uniform mixture | 6.3730 | 6.3730 |
| All pairs FNO | 17.6992 | 17.6992 |
| Fine only FNO | 23.3336 | 23.3336 |

The inverse error mixture improves on POD GP, while autoregressive POD GP and fidelity basis FNO outperform both mixtures (Table 5). Figure 4 shows that the inverse error mixture improves six evaluation cases and the fitted mixture improves five. These seven shared cases do not support a significance claim or independent training replicates.

**Why POD GP matches the training mean.** All 17 reserved POD GP predictions are identical. Their relative field difference from the independently reconstructed training mean is $5.34 \times 10^{-8}$, explaining the matching rounded errors in Table 13. The fitted coefficient GP assigns negligible covariance between the reserved and training inputs, so its predictions return to the training mean. The mixture gain therefore comes from combining this reference with the other surrogate predictions.

**Scoring convention.** We use the same relative $L^2$ metric as the PDE benchmark, evaluated on the common working grid in the supplied array representation. The original climate study instead uses cosine latitude weighted RMSE divided by the spatial standard deviation of the target (Niu et al., 2024, Appendix A.3). The NPZ files contain input and field arrays, without a column schema, calendar coordinates, or a reversible physical temperature transformation. We therefore report archive scale errors, without interpreting them as temperature errors in degrees or as area weighted global climate scores.

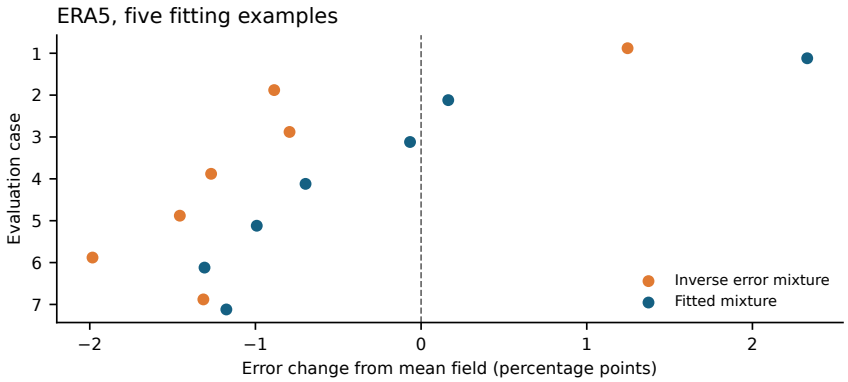


Figure 4: **ERA5 gains by evaluation case.** Change from the training mean in relative $L^2$ percentage points, averaged over three $K = 5$ fitting selections. Negative values favor the mixture. Case numbers identify archive rows rather than dates.

### E.5 Additional fitting and evaluation cases for Heat I

This separate case study generates 168 new parameter inputs from the original Heat I sampling distribution, with 40 fitting pool cases and 128 evaluation cases. None overlaps the original training or evaluation inputs. Generator replay checks reproduce the original solver outputs before creating the new fields. Neural checkpoints remain fixed. The unsaved POD GP estimator is reconstructed from original training data only.

Three fitting selections share the 128 evaluation inputs. The mixture rules and fitting budgets are fixed before scoring these cases, and each fitted weight vector is saved before evaluation answers are accessed. Table 14 reports the three main rules separately from the 22 dataset comparison. At five fitting examples, inverse error weighting reduces error relative to selection by 38.6%, and the fitted mixture by 36.9%. The campaign also evaluated other weighting rules, so this check is not a preregistered test of the final paper's full protocol. It predates the direct loss control in Appendix F.1 and tests new inputs of one PDE, rather than new problem families or training seed variability.

Table 14: **Additional Heat I inputs.** Mean relative $L^2$ errors in percent. Three fitting selections share the same 128 evaluation inputs and trained models.

| $K$ | Selected model | Inverse error mixture | Fitted mixture | All pairs FNO |
|---|---|---|---|---|
| 5 | 0.008638 | 0.005300 | 0.005454 | 0.011171 |
| 10 | 0.008638 | 0.005286 | 0.005359 | 0.011171 |
| 20 | 0.008638 | 0.005270 | 0.005296 | 0.011171 |

# F Ensemble error analysis and related methods

## F.1 Error geometry and the fitting objective

For a fixed library, Equation (6) expresses expected squared relative error as $R(w) = w^\top C w$, with $C = \mathbb{E}[G(x, y)]$. The error product matrix is uncentered, so shared bias remains part of the risk. Its diagonal describes individual accuracy and its cross terms describe error alignment (Breiman, 1996, Krogh & Vedelsby, 1994). Writing $d_m = \widehat{C}_{mm} > 0$, minimization of the diagonal approximation $\sum_m d_m w_m^2$ under simplex constraints gives $2d_m w_m = \lambda$. The scalar $\lambda$ enforces the unit sum constraint, so normalization yields $w_m \propto d_m^{-1}$. This recovers the inverse error mixture before its numerical floor, without assuming independent model errors. If some $d_m = 0$, an optimum is supported on the zero error entries.

For one case, let $e_m$ be model $m$'s normalized error field and $r_m = \|e_m\|_Q$ its magnitude. The improvement relative to errors of the same sizes that all point in one direction is

$$\left(\sum_m w_m r_m\right)^2 - w^\top G w = 2 \sum_{m<n} w_m w_n \left(r_m r_n - \langle e_m, e_n \rangle_Q\right) \geq 0. \tag{8}$$

Here $G$ is the same case's error product matrix, and $m < n$ counts each model pair once. Expansion gives the equality and Cauchy Schwarz gives nonnegativity. The identity explains how errors in different locations or directions can help either mixture. Negative correlation is not required, and the identity does not establish that a small fitting set will discover useful weights.

The fitting criterion can change the preferred mixture (Acar & Rais-Rohani, 2009, Acar, 2015). Our joint fit minimizes mean squared relative error, whereas reporting uses

$$L(w) = \mathbb{E}\left[\sqrt{w^\top G(x, y) w}\right]. \tag{9}$$

Here $L$ averages the relative $L^2$ error over inputs and targets, taking the square root separately for each field. Jensen's inequality gives $L(w) \leq \sqrt{R(w)}$ for fixed weights, but does not equate their minimizers. For example, two equally likely scalar cases with normalized model errors $e_A = (0, 2)$ and $e_B = (1.1, 1.1)$ give

$$L(t) = 1.1 - 0.1t, \qquad R(t) = 1.21 - 0.22t + 1.01t^2, \tag{10}$$

where $t \in [0, 1]$ weights model A and $1 - t$ weights B. The minimum of $L$ is at $t = 1$, while $R$ is minimized at $t = 11/101$, where $L = 110/101$. Thus fitting squared error can be about $8.9\%$ worse on the reported metric even with unlimited fitting cases. Constant spatial fields give the same example.

**Direct fitting of the reporting metric.** A standard convex alternative fits the unsquared error directly:

$$\min_{w \in \Delta_M} \frac{1}{K} \sum_{i=1}^{K} \sqrt{w^\top G_i w} = \min_{w \in \Delta_M} \frac{1}{K} \sum_{i=1}^{K} \|A_i w\|_2, \qquad A_i^\top A_i = G_i. \tag{11}$$

Here $A_i$ factors the error matrix of fitting field $i$, $K$ counts fields, and $\Delta_M$ contains nonnegative weights summing to one. Each term is a norm of a linear map. Introducing upper bounds $t_i$ with $\|A_i w\|_2 \leq t_i$ gives a second order cone problem. We evaluate this loss control on 51 partitions of the original 17 PDE tasks at $K = 5$, keeping all predictions and case assignments fixed. The weights were saved before their new scores were computed, but the predictions had previously been inspected, making this a retrospective diagnostic separate from the three main rules.

CVXPY 1.9.2 and Clarabel 0.11.1 solve the scaled problem with absolute and relative gap tolerances $10^{-10}$ and feasibility tolerance $10^{-9}$. All 51 solves returned optimal status. Eigenvalues below zero are clipped only within $10^{-12}$ of spectral scale, with larger violations rejected. The worst relative eigenvalue was approximately $-1.04 \times 10^{-18}$ and the largest simplex residual before clipping and renormalization was $4.15 \times 10^{-11}$. Source identities, reference error replay, and solver records are retained with the locked fits.

Table 15: **Fitting the reporting metric directly.** Ratios compare with the Selected model on the 17 PDE diagnostic tasks at five fitting examples. Wins and losses count datasets with lower and higher error after averaging three partitions. The direct loss fit is a supplementary control, not a fourth main rule.

| Method | Class ratio | Dataset ratio | Wins | Losses |
|---|---|---|---|---|
| Inverse error mixture | 0.882 | 0.924 | 12 | 5 |
| Fitted mixture | 0.876 | 0.918 | 12 | 5 |
| Relative $L^2$ fit | 0.859 | 0.898 | 12 | 5 |

Direct relative $L^2$ fitting reduces class aggregated error by $2.0\%$ and dataset aggregated error by $2.1\%$ relative to the original squared error fit. Error decreases on 15 datasets and increases on two. Against the Selected model, 12 improve and five worsen. The result identifies a consequential loss choice without establishing a new weighting principle or independent validation.

### F.2 Error matrix estimation and excess risk

**Proposition 1** (Error estimation and mixture risk)**.** *If* $\max_{m,n} |\widehat{C}_{mn} - C_{mn}| \leq \epsilon$ *and a feasible* $\widehat{w}$ *has empirical objective within* $\delta \geq 0$ *of the simplex minimum, then*

$$R(\widehat{w}) - R(w^*) \leq 2\epsilon + \delta, \tag{12}$$

*where* $w^*$ *minimizes population squared relative risk over the simplex.*

Here $\epsilon$ bounds matrix estimation error and $\delta$ bounds numerical optimization error. For every feasible $w$, nonnegative unit sum weights give $|w^\top(\widehat{C} - C)w| \leq \epsilon \sum_{m,n} w_m w_n = \epsilon$. Applying this once at $\widehat{w}$ and once at $w^*$, with empirical near optimality between them, proves the result. The bound separates estimation from optimization error but does not certify small $\epsilon$ from five fields. Standard concentration arguments (Hoeffding, 1963) require bounded error products and independent fitting fields, assumptions not established here. Spatial pixels are not independent fitting examples. The companion package retains the longer concentration calculation, whose bound is weaker than a trivial bounded risk guarantee at the reported budgets. Neither calculation establishes near optimality for the unsquared metric $L$.

### F.3 Comparison with prior ensemble and multifidelity methods

Model weighting, automated surrogate selection, and multifidelity field emulation are established ideas. Table 16 identifies these precedents and the scope of our integration. The contribution is the adapted portfolio, common parameter input interface, and experiments with counted fitting examples. The table compares formulations rather than measured performance. Direct comparisons with complete engineering automation systems would require matched training, tuning, and information budgets.

Table 16: **Prior systems and the scope of this study.** These workflows were not all evaluated on the present tasks. Adapted baseline implementations do not substitute for complete system comparisons.

| Predecessor | Established contribution | Present investigation |
|---|---|---|
| Engineering ensembles (Acar & Rais-Rohani, 2009, Viana et al., 2009) | Validation based weighting and model selection. | Neural field library with several fidelity strategies and counted fitting examples. |
| Functional output benchmark (Brunel et al., 2025) | Unified comparison of multifidelity field emulators. | Heterogeneous neural and reduced basis candidates combined from saved fields. |
| Stacking and AutoGluon (Wolpert, 1992, Erickson et al., 2020) | Automated library combination. | Whole field loss over a fixed portfolio of fidelity strategies. |
| EL MFS and AQBMF (Wang et al., 2024, Lee et al., 2026) | Adaptive LF fusion and MF strategy selection. | Diverse field architectures with parameter only inference. |
| MAESTRO (Kim et al., 2026) | LF and discrepancy families with screening and leave one out selection. | Direct, correction, and fine only field models. |
| LANE SI (Borisova & Nikitin, 2024) | Neural and climatological spatial forecast ensembles. | Parametric emulation with supplied multifidelity simulations. |
| HyperNOs (Ghiotto, 2026) | Automated operator training and hyperparameter search. | Fitting weights across different fidelity use strategies. |
| MFRNP and MF DeepONet (Niu et al., 2024, Howard et al., 2023) | Multifidelity field architectures. | Common ensemble fitting over multiple adapted families. |

# G REPRODUCIBILITY, AI USE, AND ETHICS STATEMENTS

## G.1 REPRODUCIBILITY STATEMENT

We will release the benchmark datasets and the code used to train and evaluate all baseline implementations, the nine surrogate models, and the ensemble methods, together with data preparation scripts, fixed splits, experiment configurations, and scripts for reproducing the reported tables and figures. Code and reproducibility materials are available at https://anonymous.4open.science/r/mf_field-35BF/README.md.

The companion package contains data preparation and generation scripts, training and evaluation implementations, fixed splits, predictions, ensemble code, and reporting scripts. Verification records reproduce individual and ensemble errors, aggregate ratios, and Elo rankings from saved predictions. Checks also cover the CPU ensemble example, campaign preparation, and a training mean control. These checks verify numerical replay and entry points. They do not constitute complete retraining of every model or regeneration of every simulation. Third party notices are retained, and redistribution of imported materials is subject to their source licenses.

## G.2 AI USE STATEMENT

Generative AI assisted coding, literature search, writing, visualization, and reference checking. The authors are responsible for the final claims, references, and artifacts, including AI assisted material.

## G.3 ETHICS STATEMENT

Inaccurate field predictions can mislead engineering or climate decisions. Benchmark error alone establishes neither physical validity nor operational suitability. Dataset construction and numerical checks are described in Appendix A, and the scope of the ERA5 metric is documented in the climate appendix. Release materials retain attribution to their sources.