Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Solver and Surrogate Choices for PDE Inference and Forecasting: A Paired Benchmark with Observation Prefixes

Jufeng Li

q3529416243@gmail.com

China University of Petroleum, East China

Wenjie Li

China University of Petroleum, East China

Junhao Zhan

China University of Petroleum, East China

Research Article

Keywords: neural operators, parameter inference, multifidelity correction, conformal prediction, computational cost, reproducible benchmarks

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11618504/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Solver and Surrogate Choices for PDE Inference and Forecasting: A Paired Benchmark with Observation Prefixes

Jufeng Li[1*], Wenjie Li[2] and Junhao Zhan[3]

[1*]College of Science, China University of Petroleum (East China), Qingdao, 266580, China.
[2]College of Mechanical and Electronic Engineering, China University of Petroleum (East China), Qingdao, 266580, China.
[3]Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, 266580, China.

*Corresponding author(s). E-mail(s): q3529416243@gmail.com;

Abstract

Partial differential equation (PDE) forecasting from sparse noisy observations requires parameter inference and future propagation. We compare pipelines using surrogate inference and forecasting (A0), surrogate inference followed by a numerical forecast (A1), and numerical prefix inference followed by a numerical forecast (P). The main benchmark covers three one-dimensional periodic PDE families, Fourier neural operator (FNO) models and U-Nets, three matched training seeds, and 200 test initial conditions per model. A1 lowers error in all 12 Allen–Cahn and Burgers configurations. A separate evaluation against a refined numerical reference reproduces this direction in all 12 configurations, with paired 95% intervals below zero. Increasing FNO training from 24 to 96 initial conditions at fixed epochs improves surrogate accuracy and narrows the additional gain from numerical forecasting. At three observed outputs, mean A1 minus A0 error changes from -0.03295 to -0.00687 for Allen–Cahn and from -0.00478 to -0.00175 for Burgers. Complete-query measurements also show substantial dependence on implementation: restricting candidate rollout to the observed prefix preserves A1 forecasts while reducing three-output query time by about 27% for Burgers/FNO and 48% for Burgers/U-Net. Direct Fourier propagation makes numerical inference cheaper than surrogate-assisted inference across the advection–diffusion horizon grid. At the operational likelihood temperatures, its forecast is more accurate at eight observed outputs and less accurate at three. Development-selected temperatures also change the P–A1 prefix comparison on a separate test block. These experiments quantify how surrogate training, inference calibration, candidate rollout length and numerical propagation affect the accuracy and cost of inference-to-forecast pipelines.

Keywords: neural operators, parameter inference, multifidelity correction, conformal prediction, computational cost, reproducible benchmarks

MSC Classification: 65M32 , 65Y20 , 68T07

1 Introduction

Forecasting a physical system from observations requires both an estimate of its unknown parameters and a model for its subsequent evolution. A learned PDE surrogate can supply both calculations. Its approximation error then enters the parameter-to-observation map used for inference and the forward map used for forecasting. Replacing either calculation with a numerical solver changes the accuracy and the work required for a complete forecast request.

The two stages create distinct opportunities for combining models. Surrogate inference can be followed by a numerical forecast at the inferred parameter, so that the numerical model acts on the prediction stage. Alternatively, a numerical solver can evaluate candidate parameters only through the observed prefix before producing the final forecast. The observation prefix determines the amount of candidate propagation required

for this second approach. Evaluating these choices together links surrogate quality, available observations and numerical cost to the forecasting task.

We compare three pipelines: surrogate inference and forecasting (A0), surrogate inference followed by one numerical trajectory at the inferred parameter (A1), and numerical prefix inference followed by one numerical trajectory (P). The study asks how forecast error, calibrated trajectory-set size and complete CPU query time change with PDE family, observation-prefix length and surrogate training scale. All pipelines start from a known initial field and receive the same noisy sensor observations within each paired instance.

This question connects several established methods. Multifidelity modelling combines models of different costs and accuracies through discrepancy corrections and other constructions (Kennedy and O'Hagan, 2000; Peherstorfer et al., 2018). Physics-based neural-operator correction has been applied to Bayesian inverse problems (Cao et al., 2023), while observation-operator design addresses the effect of model error on inference (Cvetković et al., 2024). A recent preprint compares neural operators with classical solvers for forward queries (Huang and Stuart, 2026; arXiv:2609.24021, https://arxiv.org/abs/2609.24021), and neural predictions can initialise numerical solutions (Zhou et al., 2025). Split-conformal prediction supplies calibrated uncertainty sets for surrogate outputs (Vovk et al., 2005; Gopakumar et al., 2026).

The benchmark measures the value of numerical computation at each stage through three connected experiments:

1. A matched A0/A1 comparison across three PDE families, two architectures and three training seeds, followed by a refined-reference evaluation and an FNO training-scale comparison. Together they assess whether forecast gains persist under numerical refinement and how those gains change as the surrogate improves.
2. A forecast decomposition from posterior averaging to plug-in prediction and from surrogate to numerical propagation, with an additive-discrepancy control and conformal trajectory sets.
3. A horizon experiment that measures complete query costs, restricts candidate propagation to the fields used for inference, scores a common future window, and compares RK4 with direct Fourier propagation for advection–diffusion. A separate development/test experiment measures sensitivity to likelihood temperature.

The Allen–Cahn and Burgers gains persist against refined truth and become smaller when FNO training increases from 24 to 96 initial conditions at fixed epochs. Prefix-limited surrogate inference reduces query cost at the same forecast, while direct AD propagation changes the cost ranking of the inference pipelines. The resulting benchmark relates numerical forecast gains to surrogate quality and to the work required to infer and propagate a parameter estimate.

2 Models and evaluation

2.1 PDE families and simulation reference

We use three periodic PDE families on $[0, 2\pi)$ with $N = 256$ spatial points. The production reference for the main benchmark is Fourier pseudospectral RK4 with $dt = 0.001$, producing 20 output fields at physical times $0.05, 0.10, \ldots, 1.00$.

Table 1 PDE families, inferred parameters and deployment ranges.

Family	**Equation**	**Unknown parameters**	**Deployment box**
Allen–Cahn (AC)	$u_t = \varepsilon u_{xx} + \rho(u - u^3)$	(ε, ρ)	$[0.002, 0.04] \times [0.2, 4.0]$
Burgers (BU)	$u_t + uu_x = \nu u_{xx}$	ν	$[0.02, 0.10]$
Advection–diffusion (AD)	$u_t + cu_x = \nu u_{xx}$	(c, ν)	$[0.6, 1.6] \times [0.004, 0.04]$

All pipelines receive the complete initial field. The shared initial-condition generator sums ten random-phase harmonics with amplitudes proportional to $1/m$, then centres the field and normalises it to a fixed standard deviation. The three families share their input-field construction, spatial grid and output schedule. Their parameter dimensions and dynamics give different inference and propagation tasks.

The main benchmark's reference and numerical forecasts use the same RK4 discretisation. A separate evaluation generates observations and scores forecasts against a refined reference with $N = 512$ and 100 RK4 substeps per output interval ($dt = 0.0005$). The AD solver comparison also uses direct Fourier propagation, $\widehat{u}_k(t) = \widehat{u}_k(0)\exp[-(\nu k^2 + \mathrm{i}ck)t]$, for the constant-coefficient periodic equation. Sections 4.4–4.6 compare these propagators, measure numerical refinement discrepancies and evaluate forecast accuracy against refined truth.

2.2 Observations and candidate-grid inference

Let j_{obs} denote the number of observed output fields; the physical observation time is $0.05j_{\mathrm{obs}}$. Observations are $y = Hu(1{:}j_{\mathrm{obs}}) + \eta$, where H selects equally spaced sensors and $\eta \sim \mathcal{N}(0, \sigma^2 I)$. Each initial-condition and observation-setting pair supplies one realised observation to every pipeline.

Parameter inference uses a tensor-product candidate grid $\theta_1, \ldots, \theta_K$ over the deployment box. The grid contains $K = 25$ candidates for AC and AD and $K = 13$ for BU. True parameters are drawn continuously from the box. Gaussian likelihood weights on this grid define a discrete parameter distribution, from which the pipelines form either a weighted forecast or a posterior-mean parameter estimate.

For surrogate inference, $w_k \propto \exp\left[-\frac{\|y - Hh(\theta_k)\|^2}{2\sigma_{\mathrm{eff}}^2}\right]$. The operational baseline retains the supplied development-selected likelihood temperatures $\sigma_{\mathrm{eff}} = 0.10$ for AC and 0.05 for BU and AD. Numerical prefix inference uses 0.02 in the same Gaussian weighting form, equal to the observation-noise standard deviation. Both inference paths use the same candidate support. The effective temperatures determine the concentration of the grid weights; they are prediction-risk calibration parameters. The main, horizon, refined-reference and training-scale comparisons retain these operational values.

2.3 Forecast pipelines and complete query cost

Write $h(\theta)$ for a surrogate trajectory, $G(\theta)$ for a numerical trajectory, and $\bar{\theta} = \sum_k w_k \theta_k$. The forecast pipelines and two decomposition controls are:

Table 2 Inference and propagation in the forecast pipelines and decomposition controls.

Pipeline	**Parameter estimate**	**Future trajectory**	**Numerical propagation**
A0	Surrogate weights w	$\sum_k w_k h(\theta_k)$	None
A1	Surrogate weights w, then $\bar{\theta}$	$G(\bar{\theta})$	One full trajectory
P	Numerical prefix weights w_{phys}, then $\bar{\theta}_{\mathrm{phys}}$	$G(\bar{\theta}_{\mathrm{phys}})$	K candidate prefixes and one full trajectory
A0p diagnostic	Same $\bar{\theta}$ as A1	$h(\bar{\theta})$	None
A4 control	Same $\bar{\theta}$ as A1	$\mathrm{A0} + G(\bar{\theta}) - h(\bar{\theta})$	One full trajectory

A1 requires candidate surrogate fields through j_{obs} to compute its parameter estimate, then obtains the forecast from $G(\bar{\theta})$. Its candidate rollout can consequently stop at the observation prefix. A0 uses candidate future fields to form its weighted forecast, while A4 also uses those fields in its additive correction. We measure both prefix-limited A1 and full-candidate A1 to quantify the computational effect of rollout length at a fixed forecast.

A complete query includes candidate propagation, likelihood weights, the parameter estimate and final propagation. Timings evaluate the post-training deployment calculation. Reference-data generation, observation generation, model loading, training, offline data generation and error scoring are performed outside the timed request. Each method executes independently once per initial condition and horizon, with randomised method and horizon order. Three separate initial conditions provide warm-up runs.

The timing environment is a single CPU process on Windows 11 using Python 3.12.14, NumPy 2.3.5 and PyTorch 2.15.0 CPU. PyTorch uses one compute thread and one interoperation thread; BLAS thread limits are one. Processor details, checkpoint hashes and software settings are recorded in the execution manifest. These measurements support within-environment comparisons of complete requests under a common batching configuration.

Figure 1 separates the shared observations from the inference and prediction computations. It also shows why the observed prefix limits candidate rollout for A1 and P, while A0 requires candidate future fields.

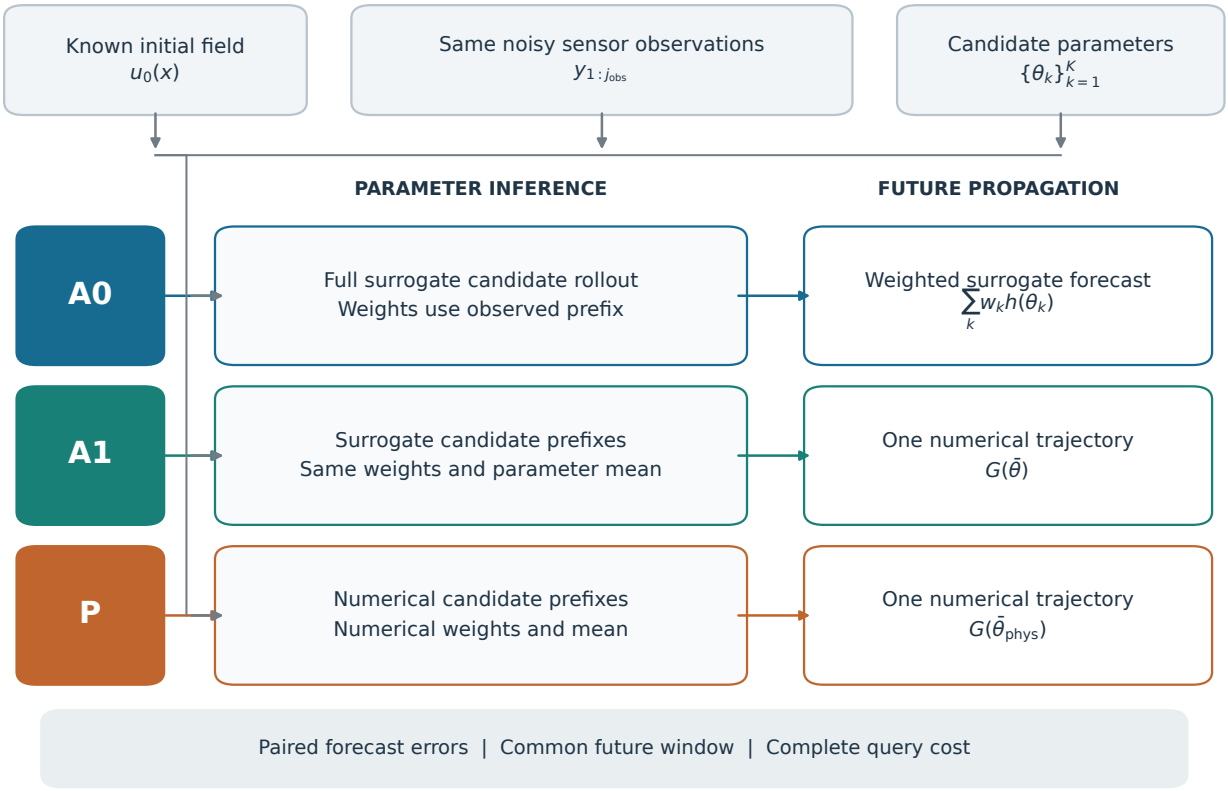


Fig. 1 Shared inputs and computational stages of A0, A1 and P. A0 uses weighted surrogate future fields; A1 and P propagate inferred mean parameters numerically. The diagram is a schematic of the implemented methods. It was drawn with Python code developed with OpenAI Codex (GPT-6-family) assistance; the method definitions and drawing source are supplied.

2.4 Error metrics and statistical units

The main comparison scores $e = \max_{j>j_{\mathrm{obs}}} \mathrm{RMS}_x(u_j - \widehat{u}_j)$ at $j_{\mathrm{obs}} = 3$ and 8. For each model and initial condition, we average the paired A1 minus A0 difference across these settings. Initial-condition bootstrap resampling produces the reported 95% percentile intervals. Main-comparison intervals use 5,000 draws and the fixed seeds in `code/analyze_step13_confirmation.py`.

The horizon experiment scores every observation prefix over the same future fields, $j = 13, \ldots, 20$. Fixing these absolute times makes the prediction target common across the five prefix lengths. A secondary score uses the full available future, $j > j_{\mathrm{obs}}$. Each initial condition receives one noise array for the longest prefix, and shorter prefixes take its leading entries. Longer prefixes therefore append measurements to the same realised noisy observations.

The horizon experiment uses 20,000 paired initial-condition bootstrap draws for mean differences and interactions. Its accuracy interaction is $[e(P) - e(A1)]_{j_{\mathrm{obs}}=8} - [e(P) - e(A1)]_{j_{\mathrm{obs}}=3}$. Positive values indicate an increase in P's relative error as the observation prefix expands. These intervals are pointwise 95% intervals for the fitted model and specified observation setting. Resampling keeps each initial condition and its repeated observation settings together. Main, horizon and refined-reference contrasts are reported separately by fitted model. The refined-reference analysis uses 20,000 draws and averages the three- and eight-output differences within each initial condition. The training-scale analysis additionally reports a two-way bootstrap over the three model seeds and 120 shared test initial conditions, with 20,000 draws. Each draw resamples both axes while preserving the paired methods and training sizes; the 360 seed-by-condition cells remain a crossed design. All intervals are unadjusted across contrasts, and an interval spanning zero leaves that contrast's direction unresolved.

2.5 Conformal trajectory sets

For each fitted model and observation setting, calibration uses the same max-time spatial-RMS error as the main comparison. With $n_{\mathrm{cal}} = 120$, nominal miscoverage $\alpha = 0.10$, and rank $k = \lceil (n_{\mathrm{cal}} + 1)(1 - \alpha) \rceil = 109$, the calibration order statistic is q. The resulting prediction set is

$$C(y) = \left\{ v : \max_{j > j_{\mathrm{obs}}} \mathrm{RMS}_x(v_j - \widehat{u}_j(y)) \le q \right\}.$$

Here q is the radius of a trajectory set in the max-time spatial-RMS metric. The CSV field `width` stores this radius. Coverage records whether the complete reference trajectory lies in $C(y)$. Under exchangeability of calibration and test instances, split conformal targets marginal trajectory-set coverage for each fixed model and observation setting. We report the radius together with realised test coverage and Clopper–Pearson intervals.

2.6 Numerical reference and training scale

The refined-reference evaluation keeps the existing checkpoints, candidate grids and operational temperatures. Periodic Fourier resampling maps each 256-point initial field to 512 points and recovers the original field at every second node. Observations come from the refined trajectory. A0's 256-point forecast is Fourier-interpolated to the fine grid, and A1 propagates its inferred parameter with the refined solver. Both are scored over outputs 13–20. Companion scores restrict the fine reference to the coarse nodes and evaluate both coarse and refined A1 forecasts there. This separates forecast-discretisation sensitivity from scoring-grid sensitivity.

The training-scale experiment fits FNO models on 24 or 96 initial conditions, using the same eight parameter draws per initial condition and the same 12-epoch recipe. The 24-condition dataset is exactly nested in the 96-condition dataset. These fits receive 3,840 or 15,360 one-step examples and 720 or 2,880 optimiser updates, respectively. Thus the comparison increases training data and optimisation together. A common held-out set evaluates A0, A1 and P at three and eight observed outputs against the production reference, using the common output window 13–20. Surrogate rollout at the true parameter measures forward approximation separately from parameter inference.

2.7 Development-set selection of likelihood temperature

We examine inference temperature separately from conformal calibration. A1 and P each select $\sigma_{\mathrm{eff}} = 0.02\gamma$ by minimising development-set mean forecast error on outputs 13–20. Selection is performed for each family and observation prefix, separately for each fitted surrogate and for the numerical estimator. The operational temperatures in Section 2.2 remain a second evaluation mode on the same test observations. These temperatures control the concentration of finite-grid likelihood weights; their selection criterion is prediction risk.

An initial grid $\gamma \in \{0.5, 0.70710678, 1, 1.41421356, 2, 4\}$ frequently selected its upper endpoint. A follow-up expands the grid with $8, 16, 32$, reuses the same development conditions, and evaluates on a fresh test block. The expanded study generates observations and numerical forecasts with the N=512, 100-substep reference. A1 evaluates surrogate candidate prefixes on N=256; P evaluates numerical candidates on N=512. The primary contrasts use the common future window. Across-seed summaries use 20,000 crossed bootstrap draws over the three fitted seeds and 100 shared test conditions, preserving the same P forecast and observation across seeds. We report the three-to-eight-output interaction alongside the selected-grid boundaries. These accuracy comparisons are separate from the complete-query timings in Section 4.4.

2.8 Use of AI-assisted research software

OpenAI Codex with GPT-6-family models assisted with supplemental experiment code, execution orchestration, numerical analysis scripts, manuscript drafting and revision, and scripted figure construction. The reported numerical results come from the saved PDE calculations and model evaluations. Raw records, source scripts, checkpoint hashes and automated validation outputs document these computations. The authors retain responsibility for the scientific interpretation, references and final submitted content. Figure captions identify AI-assisted construction, and the supplementary material provides the underlying data and derivations.

3 Experimental design

Main model comparison. The Fourier neural operator follows the spectral operator construction of Li et al. (2021). The one-dimensional U-Net adapts the encoder–decoder and skip-connection architecture of Ronneberger et al. (2015). Eighteen models cover two architectures, three training seeds and three PDE families. The matched recipe `step13_matched_v1` uses 24 training initial conditions, eight parameter draws per initial condition, 3,840 one-step pairs and 12 epochs. One-step examples share trajectories within

each training initial condition. Calibration seeds are 980000–980119; test seeds are 990000–990199. Both observation settings use 16 sensors and noise standard deviation 0.02. The data contain 4,320 calibration records and 7,200 test records.

Forecast decomposition. Seeds 995000–995024 supply 25 additional initial conditions per configuration, with model seed 0, two observation settings, two architectures and three families. This produces 300 records for the A0, A0p, A1 and A4 comparison. The decomposition measures error changes along the sequence of posterior averaging, a plug-in surrogate forecast and numerical propagation at the same parameter estimate.

Observation-horizon experiment. Initial-condition seeds start at 999200. The five prefixes are $j_{\text{obs}} \in \{2, 3, 5, 8, 12\}$, using 16 sensors and noise standard deviation 0.02. Model seed 0 is evaluated on 80 initial conditions for BU/FNO, 45 for BU/U-Net, 35 for AC/FNO and 40 for AD/FNO. The experiment contains 200 configuration–initial-condition pairs, 80 distinct shared initial-condition seeds and 4,400 method-and-horizon records. All methods within a pair share the observation; the two Burgers architectures also share their overlapping initial conditions and observations.

Refined-reference evaluation. All 18 existing models are evaluated on 90 fresh initial conditions per family, seeds 999410–999499; the ten pilot seeds 999400–999409 are excluded. There are 270 family–initial-condition pairs, 1,620 fitted-model–initial-condition pairs and 3,240 model–condition–horizon records. The 90 initial-condition seeds are shared across model configurations, with paired observations within each family and condition. Models and likelihood temperatures remain fixed.

Training-scale evaluation. Twelve new FNO fits cover AC and BU, two training sizes and model seeds 0, 1 and 2. Training seeds are 1010000–1010095, with the first 24 used by the smaller budget. Development seeds are 1020000–1020039; the 120 test seeds are 1030000–1030119. Both budgets and all model seeds share each family's test observations. The experiment contains 8,640 method–horizon records. Training wall time and peak sampled process memory are recorded per fit; query timings in this extension are secondary measurements.

The main, decomposition, horizon, refined-reference and training-scale test samples are disjoint. Protocols specify sample ranges, model choices, noise construction and scoring windows. Execution records identify checkpoint and protocol hashes. Within each experiment, pairing retains shared conditions and observations across the methods being compared.

Temperature selection. Development seeds 999600–999659 provide 60 conditions per family. The initial-grid test block is 999700–999799; the expanded-grid follow-up uses 999800–999899. All three families, both architectures, three fitted seeds and five observation prefixes are evaluated. Each family has 30 A1 model-by-prefix selections and five P selections. The expanded test contains 21,000 method-by-prefix-by-temperature-mode records: 18,000 A1 records and 3,000 shared P records. Temperature selection uses only the development block; test records from the initial and expanded grids are reported separately.

4 Results

4.1 Numerical forecasting improves Allen–Cahn and Burgers predictions

Table 3 and Figure 2 report the A1 minus A0 error averaged over the two observation settings within each of 200 test initial conditions. All 12 AC/BU configurations have negative mean differences and intervals below zero. AD varies across architectures: one U-Net configuration improves, while the other comparisons range from higher A1 error to intervals spanning zero.

The AC and BU results show that numerical propagation can improve forecasts even when the parameter estimate still comes from the surrogate likelihood. The AD pattern motivates the separate examination of posterior averaging and forecast-map replacement in Section 4.3, and of the numerical propagator in Section 4.4.

4.2 Conformal radii and realised coverage

Across the two observation settings, A1's nominal 90% conformal radius is smaller than A0's in 25 of 36 architecture, training-seed and observation-setting combinations: 11/12 for AC, 11/12 for BU and 3/12 for AD. Within those 25 combinations, the relative reduction ranges from 1.6% to 62.5%. Twenty-two of the 36 combinations decrease by more than 5%. Counts are 14/18 at $j_{\text{obs}} = 3$ and 11/18 at $j_{\text{obs}} = 8$.

A1's realised coverage ranges from 82% to 95%. Eighteen of 36 configurations fall below nominal 90%, and four individual 95% Clopper–Pearson upper bounds fall below 90%. The radius and coverage records together describe the resolution and reliability of each calibrated trajectory set. In AC and BU, numerical replacement often yields a smaller radius alongside the lower prediction errors in Table 3; application-specific assessment also needs the corresponding realised coverage.

Table 3 Mean paired A1 minus A0 error and 95% initial-condition bootstrap interval.

Architecture	Family	Seed 0	Seed 1	Seed 2
FNO	AC	−0.0260 [−0.0299, −0.0225]	−0.0296 [−0.0345, −0.0250]	−0.0186 [−0.0216, −0.0155]
FNO	BU	−0.0071 [−0.0084, −0.0060]	−0.0049 [−0.0056, −0.0043]	−0.0039 [−0.0044, −0.0034]
FNO	AD	+0.0036 [+0.0003, +0.0069]	+0.0027 [−0.0003, +0.0055]	+0.0030 [−0.0000, +0.0062]
U-Net	AC	−0.0107 [−0.0130, −0.0082]	−0.0100 [−0.0115, −0.0085]	−0.0213 [−0.0254, −0.0171]
U-Net	BU	−0.0027 [−0.0031, −0.0023]	−0.0061 [−0.0066, −0.0057]	−0.0169 [−0.0180, −0.0159]
U-Net	AD	−0.0019 [−0.0032, −0.0005]	−0.0001 [−0.0019, +0.0015]	−0.0020 [−0.0041, +0.0002]

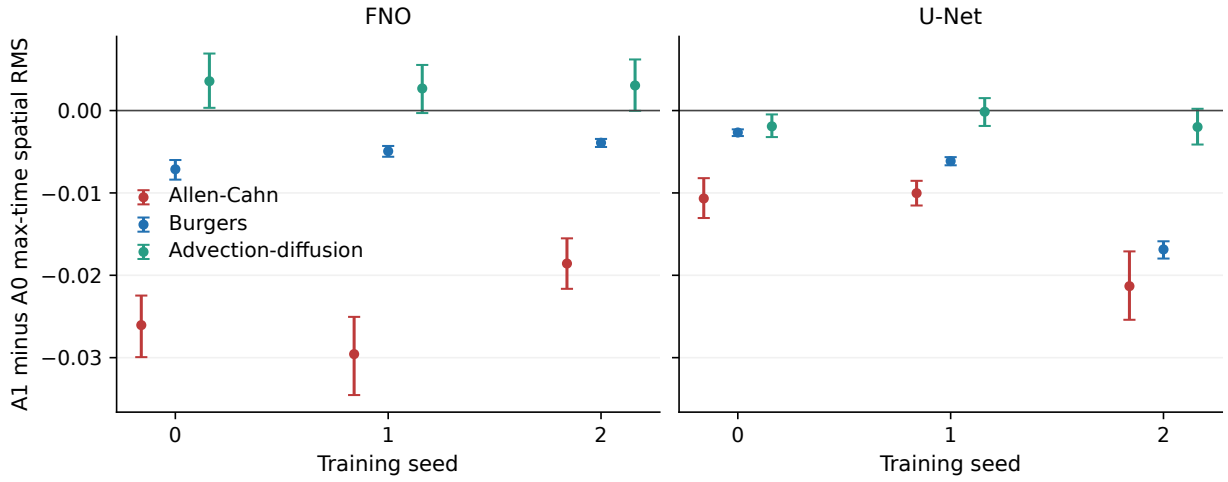


Fig. 2 A1 minus A0 trajectory error in the main model comparison. Error bars are 95% initial-condition bootstrap intervals for each fitted model. Negative values favour A1. Families are offset horizontally within each training seed for legibility. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

4.3 Posterior averaging and forecast-map replacement

Table 4 compares the forecast sequence on 25 additional initial conditions per configuration and two observation settings. The averaging difference is $e(\mathrm{A0}) - e(\mathrm{A0p})$, and the forward-map difference is $e(\mathrm{A0p}) - e(\mathrm{A1})$. Both changes use the same surrogate-derived parameter distribution. The first replaces the weighted candidate forecast with a plug-in surrogate forecast, and the second replaces that forecast's propagation model at the same $\bar{\theta}$. The trajectory differences satisfy $\mathrm{A0} - \mathrm{A1} = (\mathrm{A0} - \mathrm{A0p}) + (\mathrm{A0p} - \mathrm{A1})$.

Table 4 Mean trajectory errors and the two sequential error differences.

Architecture	Family	e(A0)	e(A0p)	e(A1)	e(A4)	Averaging difference	Forward-map difference
FNO	AC	0.0711	0.0721	0.0449	0.0451	−0.0011	+0.0272
FNO	BU	0.0108	0.0107	0.0051	0.0053	+0.0001	+0.0055
FNO	AD	0.0550	0.0543	0.0606	0.0615	+0.0006	−0.0062
U-Net	AC	0.0624	0.0616	0.0564	0.0582	+0.0008	+0.0052
U-Net	BU	0.0077	0.0075	0.0052	0.0053	+0.0002	+0.0023
U-Net	AD	0.0613	0.0614	0.0600	0.0598	−0.0001	+0.0014

The magnitude of the configuration-level mean averaging difference is at most 0.0011. Across individual records, the pooled mean absolute averaging difference is 0.0018, compared with 0.0147 for forecast-map replacement. The larger mean changes therefore occur in the propagation step. Their signs follow the family-level pattern: FNO replacement improves AC and BU, while its AD forward-map difference is −0.0062.

The two trajectory differences vary in their alignment across records. Their pooled mean absolute cosine is 0.238, with 6% of records exceeding 0.5 in absolute cosine. These instance-level statistics complement the small signed configuration means, which average over effects of both signs.

The additive control A4 has mean errors within 0.002 of A1 in all six configurations. Both use one full numerical trajectory at $\bar{\theta}$. A4 additionally evaluates $h(\bar{\theta})$ and uses the candidate future trajectories needed by A0. The comparison connects the observed gains to a standard discrepancy-correction construction and distinguishes the forecast outputs from the computations required to form them.

4.4 Candidate rollout length and numerical propagation determine query cost

Prefix-limited inference. Across all 200 configuration–initial-condition pairs and five horizons, full- and prefix-candidate A1 produce identical inferred parameters and recorded trajectory errors. The maximum recorded parameter and common-window error differences are both zero. At $j_{\mathrm{obs}} = 3$, mean query time falls from 0.1603 to 0.1171 s for BU/FNO and from 0.2387 to 0.1235 s for BU/U-Net, reductions of about 27% and 48%. Paired prefix-minus-full time differences are −0.0432 s [−0.0459, −0.0405] and −0.1151 s [−0.1182, −0.1123]. Candidate propagation beyond the observed prefix can represent a substantial share of the computation even when it leaves the final A1 forecast unchanged.

Table 5 gives the three- and eight-output comparisons; Figures 3–4 show all five horizons. Mean prefix-limited A1 time is lower than RK4 P time across all four configurations and five horizons. For Burgers, every paired P minus A1-prefix time interval is positive. At three outputs, the difference is 0.0254 s [0.0230, 0.0277] for FNO and 0.0127 s [0.0096, 0.0156] for U-Net. The increased cost of numerical likelihood evaluation is visible as longer prefixes require more candidate time steps.

Table 5 Complete mean query times and common-window mean errors. A1-full evaluates all 20 candidate output fields; A1-prefix evaluates the observed candidate prefix. Error is max-time spatial RMS over outputs 13–20.

Configuration	ICs	Observed outputs	A1-full time s	A1-prefix time s	P-RK4 time s	A1 error	P-RK4 error
BU/FNO	80	3	0.1603	0.1171	0.1425	0.00569	0.00520
BU/FNO	80	8	0.1574	0.1274	0.1984	0.00281	0.00287
BU/U-Net	45	3	0.2387	0.1235	0.1363	0.00573	0.00611
BU/U-Net	45	8	0.2392	0.1558	0.1935	0.00293	0.00314
AC/FNO	35	3	0.1994	0.1255	0.2133	0.04455	0.04370
AC/FNO	35	8	0.2021	0.1465	0.3872	0.04257	0.04551
AD/FNO	40	3	0.1903	0.1155	0.1499	0.04261	0.05312
AD/FNO	40	8	0.1885	0.1355	0.2343	0.05222	0.04993

Accuracy across observation prefixes. For Burgers, the three-to-eight-output interaction is +0.00055 [−0.00045, +0.00156] for FNO and −0.00017 [−0.00158, +0.00133] for U-Net (Table 6). Both intervals span zero. At three outputs, P minus A1-prefix error is −0.00049 [−0.00155, +0.00055] for FNO and +0.00038 [−0.00114, +0.00186] for U-Net. At two outputs, P lowers BU/FNO error, with a paired P minus A1-prefix difference of −0.00255 [−0.00420, −0.00090], and takes more computation time. The corresponding BU/U-Net error interval spans zero. The short-prefix FNO result therefore represents an accuracy–cost trade-off within this horizon grid.

Variable-window scoring produces interactions of +0.00056 [−0.00044, +0.00157] for BU/FNO and −0.00016 [−0.00159, +0.00134] for BU/U-Net. These intervals also span zero, giving the same interaction assessment for the two scoring choices. AC has an interval spanning zero. AD has a negative interaction of −0.01280 [−0.02055, −0.00601], indicating that P's relative error decreases between three- and eight-output prefixes.

Direct AD propagation. Replacing RK4 with direct Fourier propagation changes the AD cost ranking (Figure 5). At three outputs, mean complete P-analytical time is 0.000663 s, compared with 0.149909 s for P-RK4 and 0.013957 s for A1-prefix with an analytical final forecast. At eight outputs, these times are 0.001264, 0.234280 and 0.035802 s. Analytical P is cheaper than surrogate-assisted analytical forecasting

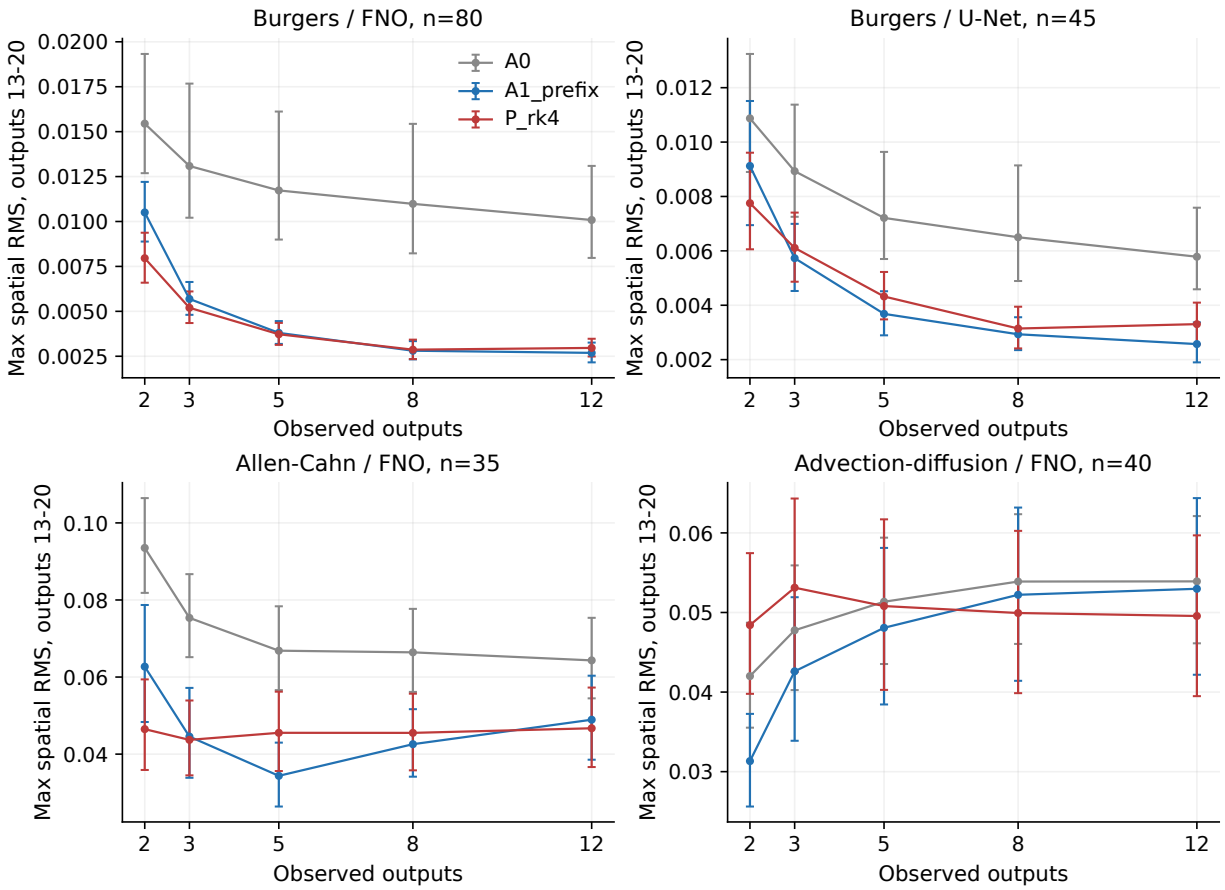


Fig. 3 Mean trajectory errors over the common output window 13–20, with 95% initial-condition bootstrap intervals. Each configuration uses one fitted model seed. Methods within a pair receive the same observation, and noise is nested across prefix lengths. Paired-difference intervals are reported in the contrast tables. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

Table 6 Paired accuracy interaction, defined as P-RK4 minus A1-prefix at eight observed outputs, minus that difference at three outputs. The common future window is outputs 13–20.

Configuration	ICs	Mean interaction	95% IC-bootstrap interval
BU/FNO	80	+0.00055	[−0.00045, +0.00156]
BU/U-Net	45	−0.00017	[−0.00158, +0.00133]
AC/FNO	35	+0.00379	[−0.01200, +0.02031]
AD/FNO	40	−0.01280	[−0.02055, −0.00601]

throughout the five tested horizons. Direct propagation reduces the cost of evaluating every candidate likelihood enough to change which inference path is computationally preferable.

The forecast errors of each inference path remain essentially unchanged by the propagator substitution. At three outputs, P-analytical minus A1-prefix-analytical error is +0.01051 [0.00459, 0.01703]. At eight outputs it is −0.00229 [−0.00443, −0.00046], with interaction −0.01280 [−0.02039, −0.00598]. Surrogate-assisted inference therefore gives the more accurate three-output forecast, while analytical P provides both lower error and lower measured time at eight outputs. These outcomes describe the candidate-grid inference paths with the likelihood temperatures in Section 2.2.

4.5 Spatial resolution and propagation accuracy

We evaluate all ten deployment-box corners for three initial conditions, seeds 999001–999003, giving 30 corner–initial-condition cases. The production solver uses $N = 256$ and 50 RK4 substeps per output interval. Temporal refinement doubles the substeps at the same grid. Spatial refinement compares $N = 256$ with $N =$

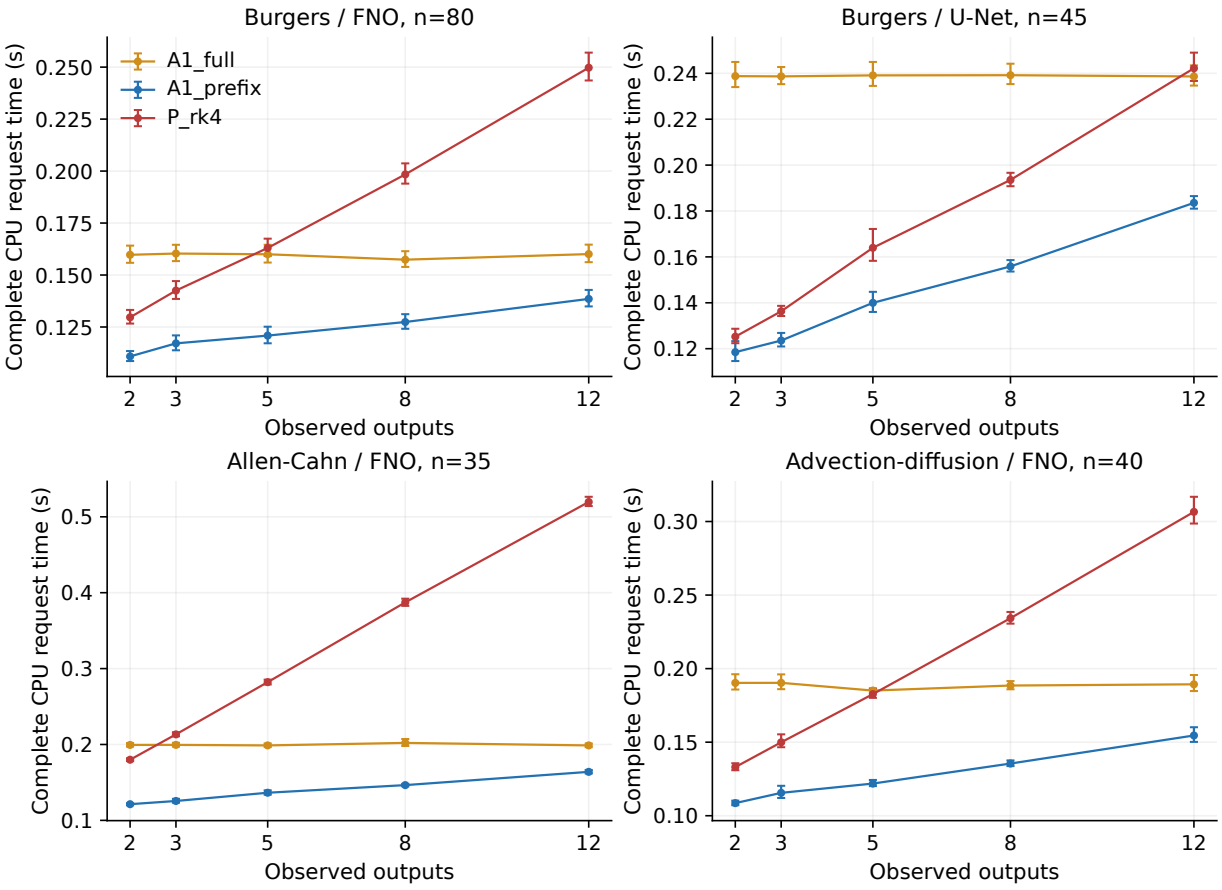


Fig. 4 Complete query times for full-candidate A1, prefix-limited A1 and RK4 P. Points are means; error bars are 95% initial-condition bootstrap intervals. Each method is timed once per initial condition and horizon under the execution protocol in Section 2.3. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

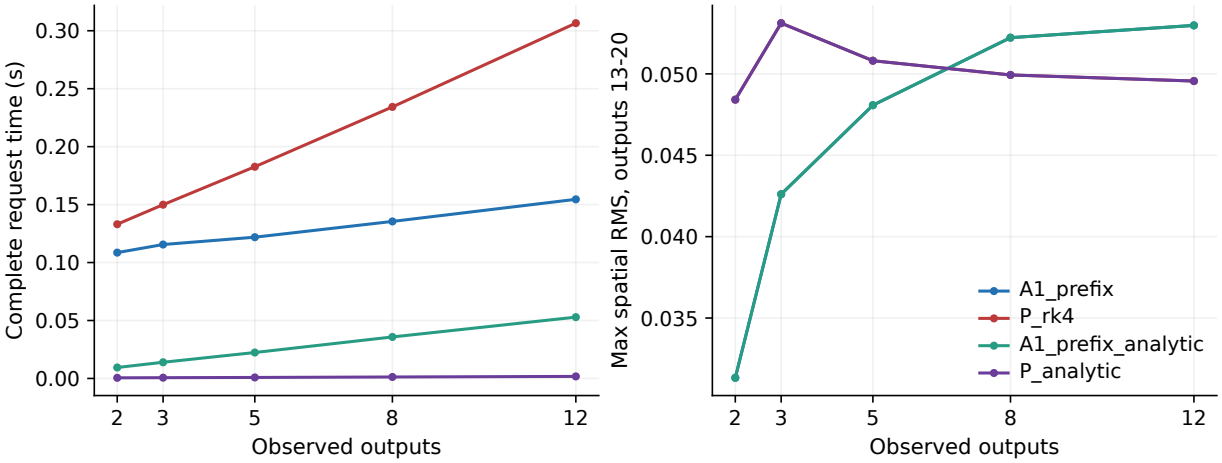


Fig. 5 AD query time and common-window trajectory error under RK4 and direct Fourier propagation. The two propagators' error curves overlap at plotting precision for each inference path. Panels show means; paired intervals appear in the contrast tables. All observations use the production RK4 reference. The analytical-truth sensitivity check gives a maximum max-time spatial-RMS difference of 2.03×10^{-10} between the two reference trajectories in this experiment. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

512 at 100 substeps each, and the combined comparison uses production against $N = 512$ with 100 substeps. Fine fields are restricted to common coarse nodes. Relative discrepancies divide the maximum-in-time spatial RMS difference by the maximum-in-time RMS amplitude of the production trajectory.

Table 7 Maximum relative discrepancies over the evaluated parameter corners and initial conditions.

Family	Corner and IC cases	Production versus N=512 and half time step	Time refinement alone	Cases exceeding 10^{-4} in the combined comparison
AC	12	3.29×10^{-3}	1.18×10^{-11}	3
BU	6	3.20×10^{-4}	8.62×10^{-11}	1
AD	12	6.01×10^{-10}	6.01×10^{-10}	0

The four cases above 10^{-4} are additionally computed at $N = 1024$ with 400 substeps per output interval. The 512-to-1024 discrepancies are 2.55×10^{-5} to 3.52×10^{-5} for the three AC cases and 3.56×10^{-7} for the BU case, normalised by the 1024-point reference amplitude. Temporal refinement changes AC and BU by less than 10^{-10} relative RMS in these cases, whereas the combined discrepancies reach 3.29×10^{-3} and 3.20×10^{-4}. The measured resolution dependence is predominantly spatial.

Direct AD propagation is also compared with production RK4 at all 25 candidate parameters for the same three initial conditions, giving 75 cases. The largest max-time spatial-RMS discrepancy is 3.52×10^{-10}. In the supplied RK4 implementation the highest spectral coefficients are frozen; direct physical propagation evolves every represented mode. For the ten-harmonic initial fields, direct propagation with and without the frozen-tail convention differs by at most 3.08×10^{-16}. This quantifies the agreement between propagators for the input fields used in the benchmark.

4.6 Forecast gains persist against the refined reference

The refined-reference evaluation reproduces the AC and BU direction across all 12 architecture–training-seed configurations (Table 8 and Figure 6). Every paired A1 minus A0 mean is negative and every corresponding 95% initial-condition interval lies below zero. Averaging the three fitted-model means gives −0.03050 for AC/FNO, −0.01315 for AC/U-Net, −0.00516 for BU/FNO and −0.00837 for BU/U-Net. These descriptive averages summarise the three fitted models; Table 8 retains their individual uncertainty.

Table 8 Paired A1 minus A0 error against the N=512 reference. Each cell gives the mean and 95% initial-condition bootstrap interval on 90 fresh conditions, averaging the three- and eight-output settings within condition. The score uses outputs 13–20.

Architecture	Family	Seed 0	Seed 1	Seed 2
FNO	AC	−0.03108 [−0.03836, −0.02452]	−0.03543 [−0.04415, −0.02805]	−0.02500 [−0.03185, −0.01897]
FNO	BU	−0.00648 [−0.00803, −0.00519]	−0.00502 [−0.00608, −0.00412]	−0.00397 [−0.00467, −0.00334]
FNO	AD	+0.00114 [−0.00320, +0.00552]	+0.00114 [−0.00269, +0.00508]	+0.00126 [−0.00265, +0.00530]
U-Net	AC	−0.00922 [−0.01296, −0.00566]	−0.01058 [−0.01294, −0.00829]	−0.01964 [−0.02539, −0.01384]
U-Net	BU	−0.00304 [−0.00355, −0.00257]	−0.00619 [−0.00673, −0.00570]	−0.01586 [−0.01733, −0.01457]
U-Net	AD	−0.00366 [−0.00534, −0.00195]	−0.00145 [−0.00349, +0.00063]	−0.00400 [−0.00664, −0.00126]

AD remains architecture-dependent. The three FNO means are positive with intervals spanning zero. All three U-Net means are negative; seeds 0 and 2 have intervals below zero, while seed 1 spans zero. The fresh sample and common scoring window therefore support the AC/BU gain without producing a uniform AD response.

The coarse and refined A1 forecast discretisations give the same signs of the mean A1 minus A0 effects when both are scored against refined truth. Across all 18 fitted models, the largest mean-effect difference

between the fine-grid score and the coarse-forecast score at restricted nodes is below 1.62×10^{-5}. The AC and BU gains are much larger than this discretisation and interpolation sensitivity on the evaluated conditions.

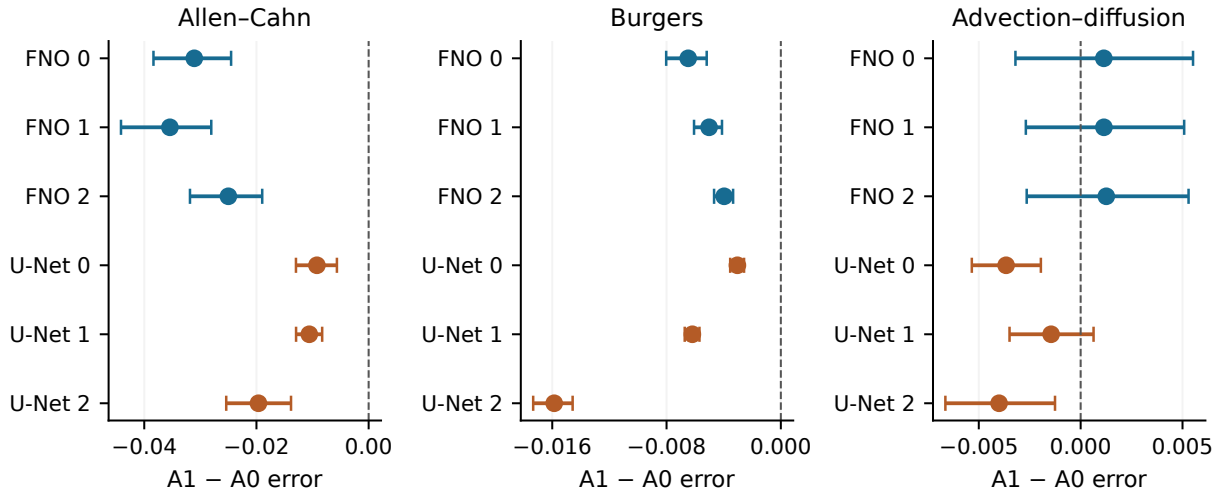


Fig. 6 A1 minus A0 common-window error against the refined reference, separately for each family, architecture and training seed. Error bars are 95% initial-condition bootstrap intervals; negative values favour A1. Each fitted model uses the same 90 condition seeds and paired three- and eight-output observations. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

4.7 Better-trained surrogates narrow the numerical forecast gain

Increasing the FNO training budget from 24 to 96 initial conditions improves forward approximation in both families. Mean true-parameter rollout error decreases by 0.05114 in AC, with a two-way bootstrap interval [0.04038, 0.06322] for the reduction, and by 0.00427 in BU, with interval [0.00335, 0.00534]. All three fitted seeds improve in each family.

A1 continues to lower the mean common-window error at the larger training budget, but its additional gain over A0 becomes smaller (Table 9 and Figure 7). At three observed outputs, AC's mean A1 minus A0 difference changes from −0.03295 to −0.00687, and BU's changes from −0.00478 to −0.00175. The corresponding changes at eight outputs are also positive. All four change intervals lie above zero, and every individual model seed has the same direction of change.

Table 9 FNO training-scale effects on common-window error. Mean effects average the three model seeds and 120 shared test conditions. Positive change means a smaller A1 gain at 96 training conditions. Intervals use the two-way bootstrap over model seeds and test conditions.

Family	Observed outputs	A1−A0, 24 ICs	A1−A0, 96 ICs	Change (96 minus 24)	95% seed-and-IC bootstrap interval
AC	3	−0.03295	−0.00687	+0.02608	[+0.01715, +0.03579]
AC	8	−0.02510	−0.00287	+0.02223	[+0.01140, +0.03378]
BU	3	−0.00478	−0.00175	+0.00304	[+0.00224, +0.00397]
BU	8	−0.00571	−0.00253	+0.00318	[+0.00249, +0.00392]

The 96-condition fits receive four times as many optimiser updates as the 24-condition fits. Training time ranges from 185.8 to 196.0 s versus 47.1 to 55.9 s in AC, and from 156.2 to 181.1 s versus 40.3 to 53.2 s in BU. The observed contraction of the A1 gain accompanies this joint increase in data and optimisation. It links the value of numerical replacement to the quality of the surrogate already available for forecasting.

As a descriptive summary of these paired contrasts, Online Resource 1 reports the retained numerical forecast gain $R = \frac{\mathrm{mean}[e(A0)-e(A1)]_{96}}{\mathrm{mean}[e(A0)-e(A1)]_{24}}$. This is a ratio of mean gains. The larger training budget retains 20.85% and 11.44% of the AC gain at three and eight outputs, respectively, and 36.52% and 44.30% of the BU gain.

The supplementary intervals preserve pairing over the three seeds and 120 shared conditions. These ratios summarise the same training-scale observations and the joint increase in data and optimiser updates.

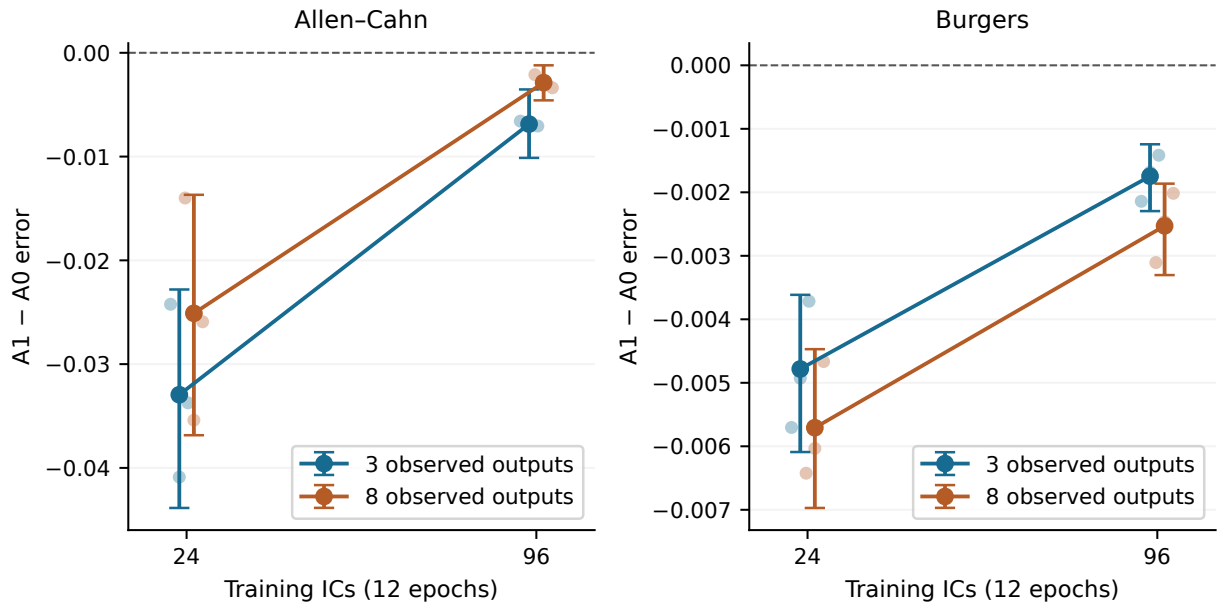


Fig. 7 Common-window A1 minus A0 error at 24 and 96 training initial conditions, for AC and BU at three and eight observed outputs. Means average three training seeds and 120 shared test conditions. Error bars use the two-way bootstrap over those crossed factors. Small translucent points show individual seed means. Negative differences favour A1; movement towards zero indicates a smaller numerical forecast gain. This figure was regenerated from the archived numerical records using Python code with OpenAI Codex (GPT-6-family) assistance.

4.8 Likelihood temperature changes the prefix comparison

The expanded-grid experiment compares operational and development-selected temperatures on the same 100 fresh test conditions per family. Table 10 reports the change in relative P versus A1 error between three and eight observed outputs. The paired comparison keeps each test observation and candidate support fixed while changing the temperature used to infer the parameter.

Table 10 Three-to-eight-output interaction under two temperature choices: $[e(P) - e(A1)]_8 - [e(P) - e(A1)]_3$. Scores use outputs 13–20. Each cell gives the mean and 95% crossed seed-and-initial-condition bootstrap interval over three fitted seeds and 100 shared test conditions.

Family	Architecture	Operational	Development-selected
AC	FNO	−0.00411 [−0.01257, +0.00485]	−0.00443 [−0.01053, +0.00084]
AC	U-Net	+0.00139 [−0.00744, +0.01087]	+0.00035 [−0.00599, +0.00749]
BU	FNO	+0.00101 [+0.00017, +0.00191]	+0.00000 [−0.00053, +0.00061]
BU	U-Net	+0.00016 [−0.00121, +0.00137]	−0.00054 [−0.00143, +0.00024]
AD	FNO	−0.00819 [−0.01345, −0.00352]	−0.00067 [−0.00444, +0.00321]
AD	U-Net	−0.00933 [−0.01449, −0.00444]	−0.00257 [−0.00691, +0.00048]

The AD interaction contracts after temperature selection: FNO changes from −0.00819 to −0.00067 and U-Net from −0.00933 to −0.00257. The short-prefix ranking also changes. For AD/FNO at three outputs, P minus A1 is +0.00863 [0.00403, 0.01354] under operational temperatures and −0.00344 [−0.00674, −0.00017]

after development selection. AD/U-Net has the same change in its mean ranking, from +0.00806 to −0.00333, with the selected-temperature interval [−0.00791, +0.00043] spanning zero. Both inference paths are retuned in these contrasts.

BU/FNO has a positive operational interaction on this fresh sample, which moves close to zero after selection. AC interaction intervals span zero under both temperature modes. All six development-selected interaction intervals span zero. Thus the operational AD short-prefix advantage and its horizon interaction depend materially on the likelihood temperatures used with the finite candidate grid.

The development choices also identify where the tested grid resolves the temperature search. AC has no endpoint selections. BU has two lower-endpoint A1 selections: FNO seed 0 at five outputs and seed 2 at three outputs. AD selects the upper endpoint at twelve outputs for all six surrogates and for P; its three- and eight-output selections are interior. The twelve-output AD results therefore describe the best value among the tested temperatures, with the upper side of the search still open. Per-model selections and five-prefix test contrasts are included in the accompanying records.

Figure 8 displays the direct three- and eight-output contrasts underlying Table 10, retaining the uncertainty for each temperature choice.

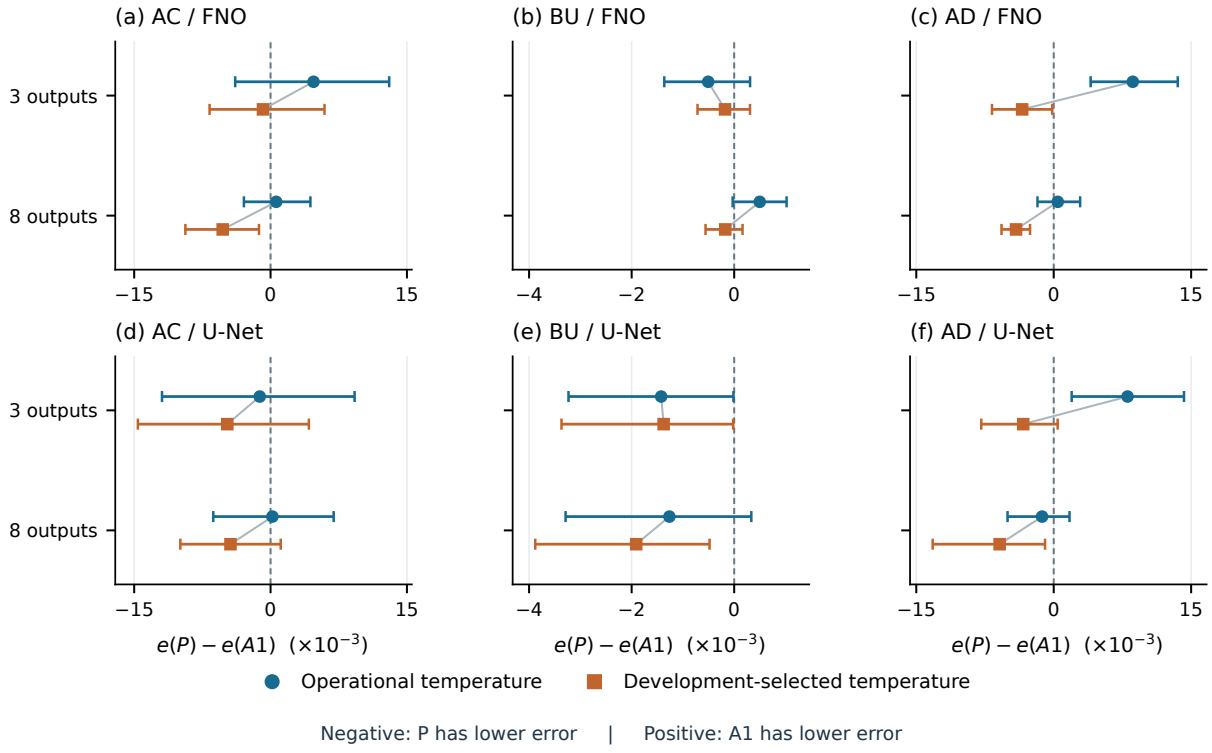


Fig. 8 P minus A1 common-window error at three and eight observed outputs under operational and development-selected temperatures. Points average three fitted seeds and 100 shared test conditions; intervals use 20,000 crossed seed-and-condition bootstrap draws. Negative values favour P. Axis limits differ across families and are shared across architectures within each family. Grey connectors pair temperature choices. Python plotting code was developed with OpenAI Codex (GPT-6-family) assistance; plotted values and their source records accompany the article.

5 Relation to existing methods

Multifidelity methods combine models with different approximation properties and costs. Kennedy and O'Hagan (2000) develop discrepancy-based modelling, and Peherstorfer et al. (2018) survey applications to uncertainty propagation, inference and optimisation. A4 evaluates an additive discrepancy at one parameter estimate. A1 uses the numerical model directly for the forecast at that estimate. The paired comparison shows how these model combinations behave when the parameter itself is inferred from a noisy observation prefix.

Cao et al. (2023) correct neural-operator predictions through a PDE-residual-based linear variational problem and study the resulting approximation and posterior accuracy in Bayesian inverse problems. The

present benchmark uses full numerical forecasts at candidate-grid parameter estimates and numerical likelihood evaluation over an observed prefix. This produces a direct comparison of computational roles at the inference and prediction stages. Huang and Perdikaris (2026) study training-free correction for neural PDE simulation, while Zhou et al. (2025) use neural-operator predictions to warm-start steady-state numerical solutions. Together, these approaches motivate evaluating the numerical work needed by the complete application.

Huang and Stuart (2026) compare neural operators with problem-matched classical solvers for post-training forward queries. Here a request first maps observations to a parameter estimate and then propagates that estimate. Prefix limitation reduces the work in the first stage, and the analytical AD comparison shows how a problem-matched propagator changes the resulting pipeline ranking.

Observation modelling is another part of this computation. Evensen (2003) develops ensemble Kalman filtering for data assimilation. Our experiment uses low-dimensional candidate-grid likelihoods with a fixed sensor layout. Cvetković et al. (2024) study observation operators that mitigate the impact of model error in Bayesian inverse problems. Varying the available prefix in our experiment changes the observations supplied to a fixed inference procedure and exposes the corresponding accuracy and cost response.

Gopakumar et al. (2026) calibrate surrogate uncertainty across output cells and architectures. We apply scalar split-conformal calibration to a trajectory-norm score. The prediction set in Section 2.5 links the forecast error metric directly to the reported radius and marginal coverage event.

6 Discussion and scope

The refined-reference and training-scale experiments explain two aspects of numerical forecast replacement. First, the AC and BU gains persist when observations and scoring use finer spatial and temporal resolution. Second, the gains contract when additional training improves FNO rollout accuracy. Numerical forecasting has the greatest measured benefit for the less accurate surrogates in this comparison. Temperature selection changes another component: the parameter estimates supplied to the final solver. In AD, development-selected temperatures reverse the mean short-prefix P–A1 ranking and reduce the horizon interaction. Prefix limitation and the analytical AD solver address how much computation inference requires before the final trajectory is generated.

The experiments cover one-dimensional periodic synthetic systems with known initial fields. The training-scale comparison covers AC and BU FNO models and changes data volume and optimiser updates together; a matched-update design would distinguish their separate contributions. The refined reference shares the production solver's spectral formulation, and higher-dimensional problems would test the pattern under different spatial structures and numerical costs. Continuous parameter inference and alternative candidate supports would extend the finite-grid construction evaluated here.

Uncertainty is reported at the level supported by each experiment. The main and refined comparisons retain results for individual fitted models, while the training-scale intervals resample three model seeds and their shared test conditions. The horizon timing study uses one fitted seed per configuration and independent CPU requests. GPU batching, compilation and reuse across repeated requests would change the computational setting. Conformal radii and coverage describe the original 24-condition models under their specified observation settings.

7 Conclusion

Numerical propagation at a surrogate-inferred parameter lowers forecast error in all 12 matched Allen–Cahn and Burgers configurations, and the same direction persists on fresh observations generated from a refined reference. Increasing FNO training from 24 to 96 initial conditions at fixed epochs substantially improves surrogate accuracy and narrows the remaining A1 gain in both families and both tested observation settings. The benefit of numerical replacement therefore depends on the surrogate quality achieved by the training budget.

Complete-query measurements show that candidate rollout length and the available propagator also matter. Prefix limitation preserves A1 forecasts and reduces three-output Burgers query time by about 27% for FNO and 48% for U-Net. With RK4 propagation, prefix-limited A1 is faster than P throughout the tested horizon grid. Direct Fourier propagation makes numerical AD inference cheaper across the grid. On fresh test conditions, development-selected temperatures change the AD short-prefix accuracy ranking, and all six selected-temperature interaction intervals span zero. The practical comparison consequently depends on surrogate training, inference temperature and the computation used at each stage.

Statements and Declarations

Funding. No funding was received to support this work.

Competing interests. The authors declare no financial or non-financial competing interests.

Author contributions. Jufeng Li: Conceptualization, Methodology, Software, Supervision, Writing – original draft. Wenjie Li: Validation, Formal analysis, Writing – review and editing. Junhao Zhan: Data curation, Visualization, Investigation, Writing – review and editing.

Data and code availability. Online Resource 2 contains the executable analysis and figure scripts, paired result records, model metadata and verification outputs supporting this article. The full reproduction archive additionally contains the model checkpoints and generated training datasets. No repository DOI is assigned in this submission package.

Ethics approval. This study uses synthetic numerical PDE simulations and involves no human participants or animals.

A Reproducibility and provenance

Paths are relative to the package root. Main-comparison records are in `results/step13_raw/`, and `code/analyze_step13_confirmation.py` computes the paired differences and 5,000-draw intervals in Table 3. The forecast decomposition uses the six per-configuration files in `results/mechanism/`, totalling 300 records. The separately archived `mechanism_all.csv` contains 50 records. Summary tables can also be regenerated with `REPRODUCE/reproduce_all.py`.

The horizon protocol is `revision/protocol.json`, and `revision/fair_benchmark.py` generates the four raw tables in `revision/results/`: `fair_bu_fno.csv`, `fair_bu_unet.csv`, `fair_ac_fno.csv` and `fair_ad_fno.csv`. The analysis entry point `revision/analyze_fair_benchmark.py` produces `fair_summary.json`, `fair_by_horizon.csv`, `fair_contrasts.csv` and `fair_interactions.csv` using 20,000 bootstrap draws. Figures are `revision/figures/fig_main_paired.png`, `fig_fair_error.png`, `fig_fair_cost.png` and `fig_fair_ad.png`.

Numerical checks are regenerated with `pythonrevision/numerical_validation.py--root.`. Outputs include `revision/results/numerical_validation.json`, `numerical_corner_convergence.csv` and `numerical_ad_validation.csv`. The production solver is `code/pdes.py`; the direct AD propagator is supplied with the numerical validation code. The execution manifest records software, processor and thread settings, checkpoint hashes and the locally frozen protocol. Archived horizon records in `results/horizon_confirm_*.csv` describe a separate experiment; Tables 5–6 use the `fair_*.csv` records listed above.

Test ranges are disjoint across the main comparison (990000–990199), decomposition (995000–995024) and horizon experiment (starting at 999200). The separately archived horizon sample uses 998000–998079. Model metadata identify architectures and training recipes. The package includes scripts for record integrity checks, numerical validation and regeneration of the reported statistics and figures.

The refined-reference protocol is `workstream/solver_protocol.json`. Its runner and analysis are `workstream/solver_validation.py` and `workstream/analyse_solver_validation.py`; verification is implemented in `workstream/verify_solver_validation.py`. The 18 raw `results/solver_{family}_{architecture}_s{seed}.csv` files and `results/solver_sensitivity_summary.json` supply Table 8. The evaluation range 999410–999499 excludes the pilot range 999400–999409. The stored summary field `configuration_ic_pairs=270` counts family–initial-condition pairs; 18 fitted models evaluated on 90 conditions give 1,620 fitted-model–initial-condition pairs.

The training-scale protocol, runner and analysis are `workstream/training_scale_protocol.json`, `workstream/train_scale_fno.py` and `workstream/analyse_training_scale.py`. Nested datasets are in `workstream/data/`, fitted checkpoints and metadata in `workstream/checkpoints/`, and per-model development and test records in `workstream/results/scale_*.csv`. `workstream/results/training_scale_summary.json` supplies Table 9, including paired size effects, per-model estimates and two-way bootstrap intervals. Data, checkpoint and source hashes are recorded alongside the run manifests. The training (1010000–1010095), development (1020000–1020039) and test (1030000–1030119) ranges are mutually disjoint and separate from the preceding experiments.

The temperature protocols are `workstream/calibration_protocol.json` and `workstream/calibration_protocol_expanded.json`. The initial and expanded reference runners are `workstream/run_calibration.py` and `workstream/run_calibration_expanded.py`. The resumable completion script is `workstream/complete_expanded_calibration.py`; it retains the completed AC files, reuses the initial-grid development scores, and computes the added temperatures and fresh heldout predictions for BU and AD. AD acceleration evaluates the amplification polynomial of the same linear RK4 discretisation, checked against the original time-stepping loop and saved development scores. It is used for the accuracy evaluation; Section 4.4

timings remain the independently measured deployment requests. `workstream/audit_completed_calibration.py` recomputes selections and recorded summaries. `workstream/analyse_calibration_crossed.py` produces the crossed-bootstrap contrasts in `workstream/results/calibration_expanded_crossed_summary.json`. The complete manifest records source, checkpoint and input hashes and the provenance of interrupted runs; the final test has 21,000 rows. `workstream/verify_q3_release.py--require-calibration` independently checks the manuscript values against these records.

References

Cao, L., O'Leary-Roseberry, T., Jha, P. K., Oden, J. T., and Ghattas, O. (2023). Residual-based error correction for neural operator accelerated infinite-dimensional Bayesian inverse problems. *Journal of Computational Physics*, 486, 112104. https://doi.org/10.1016/j.jcp.2023.112104

Cvetković, N., Lie, H. C., Bansal, H., and Veroy, K. (2024). Choosing observation operators to mitigate model error in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 12(3), 723–758. https://doi.org/10.1137/23M1602140

Evensen, G. (2003). The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 343–367. https://doi.org/10.1007/s10236-003-0036-9

Gopakumar, V., Gray, A., Oskarsson, J., Zanisi, L., Giles, D., Kusner, M. J., Pamela, S., and Deisenroth, M. P. (2026). Uncertainty quantification of surrogate models using conformal prediction. *Machine Learning: Science and Technology*, 7, 015025. https://doi.org/10.1088/2632-2153/ae2e7b

Huang, X., and Perdikaris, P. (2026). PhysicsCorrect: A training-free approach for stable neural PDE simulations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(26). https://doi.org/10.1609/aaai.v40i26.39360

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

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=c8P9NQVtmnO

Peherstorfer, B., Willcox, K., and Gunzburger, M. (2018). Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3), 550–591. https://doi.org/10.1137/16M1082469

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, 9351, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer. https://doi.org/10.1007/b106715

Zhou, X.-H., Han, J., Zafar, M. I., Wolf, E. M., Schrock, C. R., Roy, C. J., and Xiao, H. (2025). Neural operator-based super-fidelity: A warm-start approach for accelerating steady-state simulations. *Journal of Computational Physics*, 529, 113871. https://doi.org/10.1016/j.jcp.2025.113871