PINNBENCH: A Benchmark and Evaluation Study of Training Policy Selection in Hybrid PINN-Operator Solvers

Suan Lee **Namhyeon Kim** **Dongmin Jin**
School of Computer Science, Semyung University
suanlee@semyung.ac.kr

Abstract

We introduce PINNBENCH, a benchmark and evaluation study for training-policy selection in hybrid PINN-operator PDE solvers. *Headline evidence uses analytical reference solutions; a 720-run noise × sparsity sweep and d=3 and multi-scale configurations test how far the main effect survives outside that regime, and we make no claim about real-data deployment.* PINNBENCH comprises 3,545 recorded training runs across 13 PDE configurations drawn from 8 equation families, 3–5 training policies per PDE, and 10 seeds per core condition, with seven standardized metrics, Wilcoxon signed-rank tests, paired bootstrap 95% confidence intervals, Cohen's d, and Benjamini–Hochberg FDR correction. Baseline evaluation of 5 routing methods and a 2×2 stage ablation establish three findings that bear directly on how PDE solvers should be evaluated. *(i)* No single policy dominates: under PDE residual, `hybrid_full` wins 7/13, `hybrid_no_gate` 4/13, `pinn_only` 2/13. *(ii)* Accuracy and regret dissociate: under relative L_2, the physics-loss heuristic matches Random Forest and Gradient Boosting in top-1 accuracy (5/8 PDEs each)—Ridge regression reaches 75% on this objective and is not matched by the heuristic—but the heuristic suffers mean regret 0.0850 versus 0.0011 for Random Forest—a $77\times$ gap ($0.0850/0.0011$ at the four decimals of Table 7; $77.6\times$ unrounded). *Evaluation by accuracy alone is misleading for PDE policy routing. (iii)* Causal-loss effects are stage-mediated, but not through reweighting: after pretraining the causal-calibration stage lowers the PDE residual ($+5.9\%$, 6/9 significant), yet an epoch-matched stage with uniform weights does the same ($+6.0\%$); the causal weights themselves are negligible after pretraining (-0.2%) and harmful without it (-3.9%, 9/9 significant). A leave-one-family-out evaluation shows that the PDE-aware routers' 84.6% leave-one-PDE-out accuracy falls to 61.5–69.2% when a PDE's whole family is held out of training, at or below a training-free physics-loss heuristic (69.2%): part of that accuracy comes from same-family configurations. `pinn_only` shares the hybrid's reference data, so it isolates architecture, not data regime. All result JSONs, PDE implementations, policy configurations, statistical pipeline, and routing infrastructure are released as open-source artifacts.

1 Introduction

Hybrid physics-informed solvers combine PINNs [Raissi et al., 2019] with neural operators such as FNO [Li et al., 2021] and DeepONet [Lu et al., 2021], using operator pretraining to warm-start a physics-refined training pipeline [Li et al., 2024, Bischof et al., 2025]. Despite demonstrated success, practitioners face a poorly-understood design space: *which architecture variant? Should causal loss [Wang et al., 2024] be applied? Which combination should be committed to for a given PDE?*

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

Existing PDE solver benchmarks rank methods head-to-head across PDEs (e.g. PINNacle [Hao et al., 2024] compares PINN training strategies and architectures; PDEBench [Takamoto et al., 2022] compares FNO, U-Net and PINN baselines), but do not study how to *select* a training policy for a given problem within a fixed hybrid family, or how that selection should be evaluated. This paper introduces PINNBENCH, a benchmark and evaluation study that fills this gap.

Contributions. This work contributes along three axes aligned with the ED Track's scope:

1. **A controlled 13-configuration, 8-family PDE benchmark** with 3,545 recorded training runs, 10 seeds per condition, 7 standardized metrics, Wilson 95% CIs, and BH-FDR statistics (Sections 3 and 4).
2. **Baseline results for 5 routing methods** (physics-loss heuristic, Random Forest, Gradient Boosting, Ridge regression, training prior) with a full probe-then-commit protocol, leave-one-PDE-out cross-validation, and a leave-one-family-out (LOPO-F) evaluation across 8 families that retrains each router without the held-out family (Section 8).
3. **An evaluation-methodology finding:** between the Random Forest router and the physics-loss heuristic, accuracy and regret *dissociate* by $77\times$ at identical 62% top-1 accuracy. Evaluation-by-accuracy alone is misleading for PDE policy routing; regret must be reported alongside (Section 8).

What this paper is not. PINNBENCH is deliberately an *evaluation* contribution, not a new-method paper. We do not propose a new architecture, new loss, or claim state-of-the-art on any PDE. External methods (HyPINO [Bischof et al., 2025], causal PINN [Wang et al., 2024], adaptive weighting [Chen et al., 2025]) are positioned but not broadly reproduced; only internal PINO (Heat1D + AdvDiff1D, 10 seeds) is a direct same-pipeline baseline. All claims are conditioned on a synthetic regime: analytical references where they exist, PDE residual otherwise. Complementary to PINNacle [Hao et al., 2024]—which answers "which PINN variant performs best across PDEs?"—PINNBENCH answers "given a hybrid solver, which *training policy*, and how should we evaluate that choice?"

2 Related Work and Positioning

PDE benchmarks. PINNacle [Hao et al., 2024] compares $\sim$10 PINN variants spanning loss reweighting, resampling, optimizers, loss functions and architectures; PDEBench [Takamoto et al., 2022] compares FNO, U-Net and PINN baselines. Both rank methods across PDEs. PINNBENCH is complementary: it holds a hybrid solver family fixed and benchmarks *per-problem policy selection* and how that selection is evaluated. Table 1 shows where PINNBENCH sits relative to the three closest benchmarks.

Physics-informed methods, operators, and learning-curve selection. Known PINN pathologies include spectral bias [Wang et al., 2022] and convection/reaction difficulty [Krishnapriyan et al., 2021]; self-adaptive weighting [McClenny and Braga-Neto, 2023, Chen et al., 2025] adjusts coefficients online; Wang et al. [2024] propose temporal causal weighting (we show that a mean-normalized variant of it is not a reliable default in a staged pipeline). FNO [Li et al., 2021], DeepONet [Lu et al., 2021], and PINO [Li et al., 2024] learn solution maps; HyPINO [Bischof et al., 2025] combines operator pretraining with PINN refinement. Learning-curve extrapolation [Domhan et al., 2015, Mohr and van Rijn, 2024], Hyperband [Li et al., 2018], and successive halving [Jamieson and Talwalkar, 2016] underlie our multi-loss probe routing. Table 2 scopes each external method: only PINO is a direct internal same-pipeline baseline.

3 Framework and Benchmark Design

Hybrid PINN-operator architecture. Let $u_{\mathrm{PINN}}(t, \mathbf{x})$ and $u_{\mathrm{OP}}(t, \mathbf{x})$ denote the PINN and operator predictions and let $g(t, \mathbf{x}) \in [0, 1]$ be the causal-gate output. The final prediction is

$$u(t, \mathbf{x}) \;=\; g(t, \mathbf{x})\, u_{\mathrm{OP}}(t, \mathbf{x}) + \big(1 - g(t, \mathbf{x})\big)\, u_{\mathrm{PINN}}(t, \mathbf{x}), \tag{1}$$

matching the repository implementation (`src/causal_pinn/models/hybrid/causal_gate.py`: `gate * operator_pred + (1 - gate) * pinn_pred`); $g \to 1$ reproduces the operator and $g \to 0$ the PINN.

Table 1: PINNBENCH vs adjacent benchmarks. Per-problem policy selection and its evaluation (rows 6–8) appear in none of PINNacle, PDEBench, or HyPINO; PINNacle does compare training strategies, but ranks them rather than selecting among them. Entries for prior benchmarks are taken from their published versions (PINNacle: NeurIPS 2024 D&B; PDEBench: NeurIPS 2022 D&B; HyPINO: NeurIPS 2025) and were re-checked against those texts.

Property	PINNBENCH (this)	PINNacle	PDEBench	HyPINO
Primary axis	Training-policy selection	PINN variants (train. + arch.)	Arch. + dataset	Multi-physics op.
#PDE configs	13 (8 families)	22 tasks (1D–ND)	11 PDEs	7 in eval suite
#Methods compared	5 routing methods	$\sim$10 PINN variants	3 baselines	4 (w/ U-Net, Poseidon, PINO)
Seeds per cell	10	3	unspecified	not reported
Statistical pipeline	BH-FDR + Wilson + boot.	mean $\pm$ std	mean $\pm$ std	point estimates
Routing eval (LOPO + boot.)	yes	absent	absent	absent
Family holdout (LOPO-F)	yes	absent	absent	absent
Causal $\times$ pretrain	yes (2$\times$2, 9 PDE)	absent	absent	absent

Table 2: External baseline positioning. No broad head-to-head SOTA claim is made. "Contextual probe" means the method is executed under its own pipeline for positioning, not re-run in our pipeline.

Method	Type	Scope here	What is compared	Direct H2H?
PINN-only (ours)	Baseline	Same pipeline, same supervised data	All three axes	Yes
PINO [Li et al., 2024]	Operator	Heat1D + AdvDiff1D, 10 seeds	Rel. L_2, residual	Yes
HyPINO [Bischof et al., 2025]	Hybrid operator	Native suite + 2-PDE adaptation (3 seeds)	Smoke / adapter	Contextual probe
Causal PINN [Wang et al., 2024]	PINN variant	4-PDE single-stage probe	Rel. L_2, residual	Contextual probe
Adaptive wt. [Chen et al., 2025]	Weighting	Cited; not reproduced	Contextual only	Contextual only

Three-stage training pipeline. *Stage 1 (Pretraining)* trains all parameters for 200 epochs on $\mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{bc/ic}}$, where $\mathcal{L}_{\text{data}}$ fits samples of the exact solution; on the 5 configurations without one (Burgers1D $\times 3$, ReactDiff1D $\times 2$) it is a physics-only warm-up on $\mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{bc/ic}}$. *Stage 2 (Physics Refinement)* fine-tunes with $\mathcal{L}_{\text{total}} = \lambda_{\text{data}}\mathcal{L}_{\text{data}} + \lambda_{\text{physics}}\mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{bc/ic}}$. *Stage 3 (Causal Calibration)* replaces $\mathcal{L}_{\text{PDE}}$ with a causal loss adapted from Wang et al. [2024] (who use the *sum* of earlier residuals, $\epsilon \leq 10^2$; we use the *mean*):

$$\mathcal{L}_{\text{causal}} = \frac{1}{N_c}\sum_{k=1}^{K} w_k \sum_{j \in I_k} \left|\mathcal{N}[\hat{u}](\mathbf{x}_j, t_j)\right|^2, \quad w_k = \exp\Big(-\epsilon \, \text{mean}_{j \in I_1 \cup \cdots \cup I_{k-1}} \left|\mathcal{N}[\hat{u}](\mathbf{x}_j, t_j)\right|^2\Big), \quad (2)$$

i.e. the weight of time bin k decays with the mean squared residual over all earlier bins, with $K{=}100$ bins and ϵ annealed linearly from 0.01 to 1 over the 200 Stage-3 epochs.

PDE suite (13 configurations, 8 families). Table 3 summarizes the suite spanning diffusion, advection, shocks, dispersion, interface dynamics, reaction, and 2D wave/flow. Full definitions (domains, boundary conditions, exact-solution source) are in Appendix A.

4 Experimental Protocol

Policies. Policies (3–5 per PDE): `pinn_only` (PINN backbone under the same Stage 1 supervised-exact-solution data channel and epoch budget as the hybrids—an architectural control, *not* a standard data-free PINN), `hybrid_no_gate` (operator pretraining + PINN losses without the learned gate), `hybrid_full` (with gate; $\lambda_{\text{physics}}{=}1$), and `hybrid_full_lambda0.1` / `hybrid_full_lambda0.5` (the same with $\lambda_{\text{physics}}{=}0.1$ / 0.5; on Heat1D, AdvDiff1D, AllenCahn1D and two Burgers configurations).

Training schedule. In the 13-PDE suite and the stage ablation: FNO modes=8, width=32, 4 spectral layers; PINN backbone 4×64 GELU (the single-PDE studies of Section 5 use 3 hidden

Table 3: PDE benchmark suite: 13 configurations across 8 families.

Configuration	Equation	Parameters	Regime
Heat1D	$u_t = \alpha u_{xx}$	α=1	Smooth diffusion
AdvDiff1D	$u_t + cu_x = \alpha u_{xx}$	c=1, α=0.1	Advection–diffusion
AdvDiff1D (high Pe)	$u_t + cu_x = \alpha u_{xx}$	c=5, α=0.01	Sharp transport
Burgers1D	$u_t + uu_x = \nu u_{xx}$	ν=0.01	Shock formation
Burgers1D (low ν)	$u_t + uu_x = \nu u_{xx}$	ν=10^{-3}	Sharp shock
Burgers1D (very low)	$u_t + uu_x = \nu u_{xx}$	ν=10^{-4}	Very sharp shock
KdV1D	$u_t + uu_x + \delta u_{xxx} = 0$	δ=1, amplitude 3	Dispersive soliton
AllenCahn1D	$u_t = \varepsilon^2 u_{xx} + u - u^3$	ε=0.01	Interface
AllenCahn1D (sharp)	$u_t = \varepsilon^2 u_{xx} + u - u^3$	ε=0.001	Sharp interface
Wave2D	$u_{tt} = c^2 \nabla^2 u$	c=1	Wave propagation
ReactDiff1D	$u_t = Du_{xx} + ru(1-u)$	D=0.01, r=1	Moderate reaction
ReactDiff1D (stiff)	$u_t = Du_{xx} + ru(1-u)$	D=0.01, r=10	Stiff reaction
NavierStokes2D	$\omega_t + \mathbf{v}\cdot\nabla\omega = \nu\nabla^2\omega$	ν=0.1	Taylor–Green vorticity

layers of 64 and 3 FNO layers); stage epochs $200/500/200$; Adam with LR $10^{-3}/10^{-4}/10^{-5}$; 4,000 collocation and 500 boundary points; batch size 256. Core experiments use 10 seeds $\{42, 123, 456, 789, 1024, 2048, 3072, 4096, 5120, 6144\}$; secondary studies ($\lambda$-sweep, OOD, robustness grid, adaptive ϵ, external probes) use 3–5 seeds and the mechanism diagnostics one, as documented in Appendix L (Table 26). Hardware: 4×NVIDIA TITAN Xp (12 GB), PyTorch 2.5.1+cu121, Python 3.12.

Metrics. Relative L_2 error (solution accuracy) and PDE residual norm (physics compliance) are the primary metrics, with MSE, MAE, max error, boundary-condition violation, and initial-condition violation as secondary (Appendix C).

Statistical methodology. The tested paired comparisons (Table 4(a,b), Table 5, Table 11) use (i) exact Wilcoxon signed-rank tests on paired seeds, (ii) bootstrap 95% confidence intervals for paired mean differences, and (iii) paired Cohen's d. Benjamini–Hochberg FDR correction is applied at q=0.05 within each family of tests (the 13 core comparisons; the 27 stage-ablation tests).

Fairness and data accounting. Comparisons control for training schedule and supervised sample access within the synthetic exact-solution regime; Stage 1 reference-solution cost is excluded and external methods are not tuned to their own best schedule. Reference data is *not* confined to Stage 1: whenever an analytical solution exists, Stage 2 also carries a supervised term $\lambda_{\text{data}}\mathcal{L}_{\text{data}}$ for the hybrid and `pinn_only` alike. What distinguishes Stage 1 is that it is a separate 200-epoch warm-up at the highest learning rate before physics refinement, so the pretraining contrast throughout is about *when* and how long the model is warmed up, not whether a policy has the data channel.

5 Does Hybridization Help? In-Domain, OOD, and PINO

`pinn_only` disclosure. `pinn_only` gets the same reference-data channel as the hybrids (Section 4): it isolates the operator branch under matched data access and is not a data-free PINN.

Evidence matrix. Table 4 combines in-domain, PINO and OOD evidence.

Bolding in panel (a) compares `pinn_only` with Hybrid (full) only; the "Note" column reports the no-gate variant. On Heat1D the no-gate variant (0.021 ± 0.007) is in fact slightly better than Hybrid (full) (0.024 ± 0.010); both vastly outperform `pinn_only` (0.402), so the architectural finding is unaffected, but the bolded 0.024 should not be read as the column-best across hybrids.

The comparison to HyPINO [Bischof et al., 2025] needs similar care. Its adapter reaches rel. L_2 of 0.0115 on Heat1D after 500 Adam steps of fine-tuning from a pretrained foundation checkpoint (Appendix F, Table 16), whereas our hybrid trains from scratch over $200/500/200$ epochs. The two are not directly budget-comparable and we make no claim to beat HyPINO on Heat1D. Under a matched wall-clock budget the from-scratch hybrid loses on both Heat1D and AdvDiff1D (Appendix F.1).

Table 4: Main evidence matrix: (a) in-domain hybrid vs. `pinn_only`, (b) internal PINO, (c) OOD degradation (Mult. = OOD/in-domain rel. L_2 of the same model). Lower is better; p is BH-FDR adjusted; bolding in (a) compares `pinn_only` with Hybrid (full) only.

(a) In-domain hybrid vs. `pinn_only` (matched data channel), 10 seeds (Rel. L_2)

PDE	`pinn_only`	Hybrid (full)	Cohen's d	p (FDR)	Note
Heat1D	0.402 ± 0.048	$\mathbf{0.024 \pm 0.010}$	-8.33	0.004	no-gate 0.021 ± 0.007
Wave2D	0.708 ± 0.010	$\mathbf{0.434 \pm 0.078}$	-3.55	0.004	residual ↑
KdV1D	0.086 ± 0.217	$\mathbf{0.003 \pm 0.002}$	-0.38	0.006	high PINN variance
AllenCahn1D	0.226 ± 0.065	$\mathbf{0.025 \pm 0.013}$	-3.12	0.004	—

(b) Internal PINO comparison, 10 seeds (Rel. L_2 / PDE residual)

PDE	`pinn_only`	PINO	Hybrid (full)	Cohen's d (hybrid vs. PINO)	
Heat1D	$0.397/0.062$	$0.763/\mathbf{0.007}$	$\mathbf{0.018}/0.010$	-41.3 (acc.)	PINO p=0.004
AdvDiff1D	$0.587/0.090$	$0.035/\mathbf{0.009}$	$\mathbf{0.018}/0.020$	-1.40 (acc.)	PINO p=0.006

(c) OOD taxonomy, mean Rel. L_2 error; Mult. = OOD/in-domain

Shift	`pinn_only` (in/OOD)	Mult.	Hybrid full (in/OOD)	Mult.	
Wave2D speed	0.708 / 1.041	$\mathbf{1.5}\times$	0.434 / 0.981	$2.3\times$	10 seeds
Heat1D diffusivity	0.389 / 0.811	$\mathbf{2.1}\times$	0.018 / 0.441	$24.4\times$	5 seeds
AdvDiff1D velocity	0.580 / 0.909	$\mathbf{1.6}\times$	0.017 / 0.701	$40.7\times$	5 seeds
Wave2D domain $\times$1.5–2	0.706 / 0.726	$\mathbf{1.0}\times$	0.419 / 1.467	$3.5\times$	5 seeds
Wave2D domain $\times$0.8/1.3	0.706 / 0.804	$\mathbf{1.1}\times$	0.419 / 0.791	$1.9\times$	5 seeds

This is consistent with pretraining mattering, but HyPINO's multi-PDE foundation pretraining is not our per-PDE Stage 1, so it does not test our claim.

In-domain: the operator branch helps under matched data access. All four PDEs show significant in-domain hybrid improvements, large on three (Cohen's $d = -3.1$ to -8.3) and small on KdV1D ($d = -0.38$, where `pinn_only` has one outlying seed). All 4 in-domain hybrid-vs-`pinn_only` comparisons in Table 11 are BH-FDR-significant at q=0.05, as are the Wave2D OOD row and the 2 PINO accuracy comparisons; 3/5 causal-loss effect rows also pass (two improvements and one degradation, Burgers1D). The no-gate Heat1D ablation (0.021 ± 0.007 vs. 0.024 ± 0.010 with gate; PINN 0.402 ± 0.048) shows that the gate is not what carries the accuracy gain.

PINO: accuracy–residual tradeoff. PINO reaches the best residual on Heat1D (0.007) but worst L_2 (0.763); the hybrid reaches much better L_2 with a modest residual concession. This positions PINO as a limited internal operator baseline, not as a claim about PINO's behavior in general.

OOD: hybridization does not solve robustness. Under five coefficient and domain shifts, the hybrid's proportional degradation is larger than `pinn_only`'s in every case, and on the 1.5–2$\times$ domain shift its absolute error is twice `pinn_only`'s. On Wave2D coefficient shift the hybrid keeps a small but significant OOD advantage (10 seeds: 0.981 vs. 1.041, $d = -2.58$, $p_{\text{adj}} = 0.003$). The operator branch helps interpolation, not robustness.

6 Causal Loss: Extra Training, Not Reweighting

2$\times$2 factorial ablation. We cross two binary factors—*skip_stage1* (Stage-1 pretraining) and *skip_stage3* (the 200-epoch causal-calibration stage)—on 9 PDEs with 10 seeds (360 runs). Table 5 (heatmap in Figure 3, Appendix G) shows that *with* pretraining the causal-calibration stage lowers the PDE residual on all 9 PDEs ($+5.9\%$, 6/9 significant after BH-FDR) and *without* it the mean change is -2.4% (5/9 worse, 3 significantly).

Epoch-matched controls. In the 2$\times$2 design each causal arm trains 200 more epochs and each no-pretraining arm 200 fewer; two controls (360 runs, Table 21) remove these confounds. A Stage 3 with uniform weights (ϵ=0, same epochs and loss terms) gives the same gain after pretraining ($+6.0\%$, 6/9 significant); relative to it the causal weights change the residual by -0.2% (all within $\pm 1.2\%$) after pretraining and by -3.9% without it (worse on 9/9, all significant). The gain is extra training, not reweighting. With total epochs matched, removing Stage 1 still raises the residual by 125–694% (9/9 significant).

Table 5: 2×2 stage ablation (PDE residual, 10 seeds). $\Delta\% =$ (no-causal $-$ causal)/no-causal, mean with paired-bootstrap 95% CI; ✓/× significant improvement/degradation, $\approx$ not significant (exact Wilcoxon, BH-FDR over 27 tests). Causal arms also train 200 more epochs; Table 21 gives the epoch-matched controls. From `causal_ablation_table.py`.

	With Pretrain (Stage 1)		**Without Pretrain (Stage 1)**	
PDE	$\Delta\%$	Verdict	$\Delta\%$	Verdict
Heat1D	$+8.8\ [+5.5, +13.0]$	✓	$-4.1\ [-7.7, -0.1]$	$\approx$
AdvDiff1D	$+8.9\ [+3.1, +14.4]$	✓	$-13.3\ [-16.8, -9.4]$	×
Burgers1D (ν=0.01)	$+2.9\ [-0.6, +6.1]$	$\approx$	$-9.3\ [-11.4, -6.9]$	×
Burgers1D (ν=10^{-4})	$+2.0\ [-2.6, +6.3]$	$\approx$	$+1.6\ [+0.2, +3.0]$	$\approx$
KdV1D	$+7.6\ [+5.3, +10.0]$	✓	$-2.0\ [-4.7, +0.9]$	$\approx$
AllenCahn1D (ε=0.01)	$+5.7\ [+1.4, +10.1]$	$\approx$	$+5.4\ [+3.2, +7.7]$	✓
AllenCahn1D (ε=0.001)	$+4.9\ [+1.3, +8.7]$	✓	$+5.2\ [+3.1, +7.4]$	✓
ReactDiff1D (stiff)	$+7.2\ [+3.3, +11.4]$	✓	$-7.1\ [-11.5, -3.4]$	×
NavierStokes2D	$+4.8\ [+2.6, +6.7]$	✓	$+1.9\ [+0.8, +3.0]$	✓
Mean	$+5.9$	6/9 ✓	-2.4	3/9 ×, 3/9 ✓

Mechanism: what the diagnostics show. We test two mechanistic hypotheses (Appendix G; single seed each). (i) Cosine similarity between $\nabla_\theta \mathcal{L}_{\text{causal}}$ and $\nabla_\theta \mathcal{L}_{\text{PDE}}$ stays at or above 0.997 (mean ≥ 0.999) on 4 PDEs (Heat1D, AdvDiff1D, KdV1D, sharp AllenCahn1D; per-pair min/max/mean in Appendix G.1), which argues against gradient conflict on these runs. (ii) At a diagnostic ϵ=100 on Heat1D, the mean weight over the 100 bins is 0.29 without pretraining and 0.89 with it; at the training $\epsilon \leq 1$ final minimum weights stay ≥ 0.94 and are lower *with* pretraining on 3 of 4 PDEs (Table 24), so weight collapse is not demonstrated; this fits the near-zero reweighting effect after pretraining.

Practical guideline. Pretrain, then train briefly at a low learning rate; causal weights add nothing after pretraining and hurt without it, and an adaptive ϵ does not replace pretraining (Appendix H).

7 Robustness Outside the Idealized Regime

Noise $\times$ sparsity. To test how far results measured against clean analytical references carry, we cross relative Gaussian noise on the Stage-1 reference, $\sigma \in \{0, 0.05, 0.2\}$, with the number of reference points, $n \in \{1024, 256, 64, 16\}$ (16 is a 64-fold cut). Burgers1D (Cole–Hopf, ν=0.01), ReactDiff1D (Fisher–KPP travelling wave, r=1) and KdV1D (unit-amplitude soliton) are exact-solution variants, not the configurations of Table 3. Each cell runs with and without pretraining over 5 seeds—120 runs per PDE, 720 over six equation families. These probes use a lighter, operator-free pipeline in which both arms see the same (noisy, subsampled) reference and differ only by the Stage-1 phase: a 4×64 PINN whose "pretraining" is a 200-epoch fit to the (noisy, subsampled) reference, followed by 400 physics epochs with the same data term (λ_{data}=10) and 200 causal epochs. They therefore test the *warm-start* part of the pretraining effect, not the operator branch.

The direction is invariant; the magnitude is not. Pretraining wins in all 72 cells (Table 6), at $1.7\times$ to $20\times$, so the magnitude should never be quoted as one number. Noise alone is close to free: Heat1D moves between $10.2\times$ and $10.3\times$ from σ=0 to 0.2 at full density. Two problems lose a large part of their advantage when scarcity *and* noise coincide—KdV ($18.9\times \to 3.0\times$, -84%) and Fisher–KPP ($15.4\times \to 8.7\times$, -43%)—while tolerating either alone. Both carry a *travelling* localised structure; Allen–Cahn, whose front is localised but stationary, does not collapse ($1.1\times$). We read this as a hypothesis consistent with six problems, not a demonstrated mechanism: locating a moving feature from a few noisy samples is where the reference channel stops helping.

Dimension and scale. At d=3 (Heat3D, separable exact solution) the effect survives but narrows: 0.467 ± 0.041 with pretraining against 0.658 ± 0.011 without over 10 seeds ($1.41\times$, ranges disjoint), against 8.4–$10.5\times$ for the same equation in 1D. MultiscaleHeat2D superposes two Dirichlet eigenmodes whose wavelengths differ $4\times$ and decay rates $16\times$, so it stays analytic while early and late dynamics are dominated by different structure. The headline contrast survives in aggregate

Table 6: Pretraining advantage (rel. L_2 without pretraining over with) at the corners of the noise × sparsity grid; dense n=1024, sparse n=16, clean σ=0, noisy σ=0.2; collapse = easiest corner / hardest. Heat3D (d=3) and MultiscaleHeat2D use the same operator-free pipeline without the causal stage (200 + 400 epochs), 10 seeds. Every grid cell exceeds 1; the collapse column is a ratio of cells.

PDE	clean+dense	clean+sparse	noisy+dense	noisy+sparse	collapse
Heat1D	10.2×	8.4×	10.3×	8.6×	1.2×
AdvDiff1D	3.7×	2.2×	4.2×	2.3×	1.6×
Burgers1D (shock)	1.7×	1.7×	2.2×	2.1×	0.8×
KdV1D (soliton)	18.9×	14.4×	17.8×	**3.0**×	**6.3**×
AllenCahn1D (static front)	2.5×	2.5×	2.5×	2.3×	1.1×
ReactDiff1D (moving front)	15.4×	14.2×	15.4×	**8.7**×	1.8×
Heat3D (d=3)	1.41×	—	—	—	—
MultiscaleHeat2D	2.68×	—	—	—	—

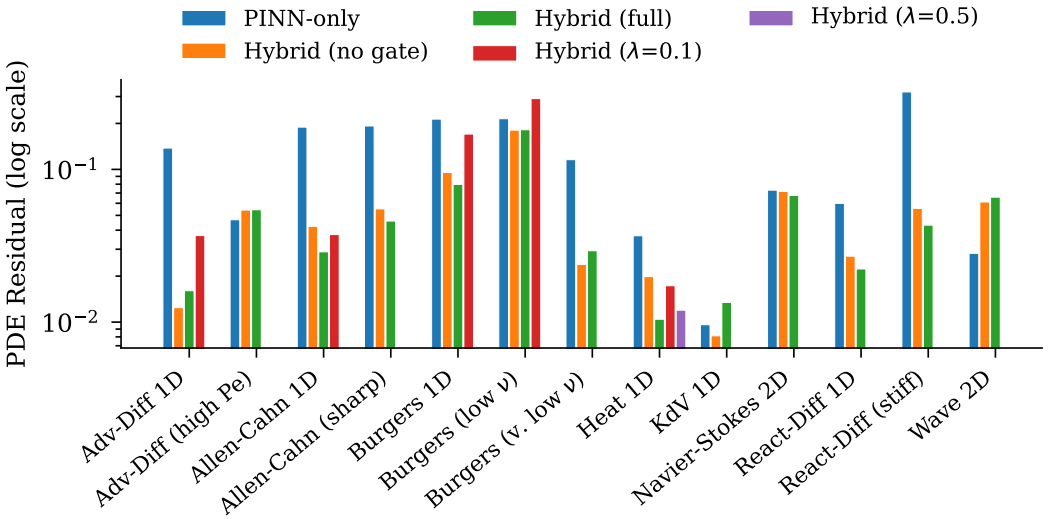


Figure 1: No-dominance: best-performing policy varies across 13 PDE configurations under the PDE-residual objective; no single policy wins everywhere. This motivates automatic routing.

(0.304 ± 0.011 vs 0.814 ± 0.008, 2.68×, 10 seeds), but projecting each learned field onto the two modes shows where it comes from: the coarse coefficient (exact 1.0) is recovered as 0.853 with pretraining and 0.478 without, the fine one (exact 0.5) as 0.004 and 0.0002. *Neither branch learns the fine scale.* Running five policies of the main pipeline on this PDE (`pinn_only`, `hybrid_no_gate`, `hybrid_full`, `hybrid_full` with λ_{physics}=10, and PINO; 200/500/200 epochs, 50 runs) does not change that: no run recovers more than 4.4% of the fine coefficient, and the four operator-bearing policies lie within 1.6% of each other in relative L_2, so the worst available choice costs a regret of only 0.06. The fine-scale failure belongs to the whole family and is not something a router could avoid. Our pretraining result is therefore a statement about the *resolved* scale.

What these probes cannot reach. Every probe above is scored against an analytical solution, because regret needs an oracle. For field data without one, the PDE residual (already one of our seven metrics) can stand in as a regret surrogate; it is a weaker instrument, and it is exactly where we would expect the invariance above to be weakest.

8 Routing and Policy Selection

Three routing variants are reported and should not be conflated: **Residual routing** predicts the policy that minimizes PDE residual at test time (Table 7 left; LOPO across 13 PDEs, 450 samples). L_2 **routing** predicts the policy that minimizes relative L_2 at test time (Table 7 right; LOPO across 8 PDEs

Table 7: Routing accuracy and regret under both objectives: residual over 13 PDEs (450 samples), L_2 over the 8 with exact solutions (280). Regret is absolute (chosen − oracle mean metric) except entries marked †, which are relative to the oracle. 84.6% uses fold-wise top-5 feature selection; k and the variant were chosen on these same folds, so it is optimistic and is not the default RF.

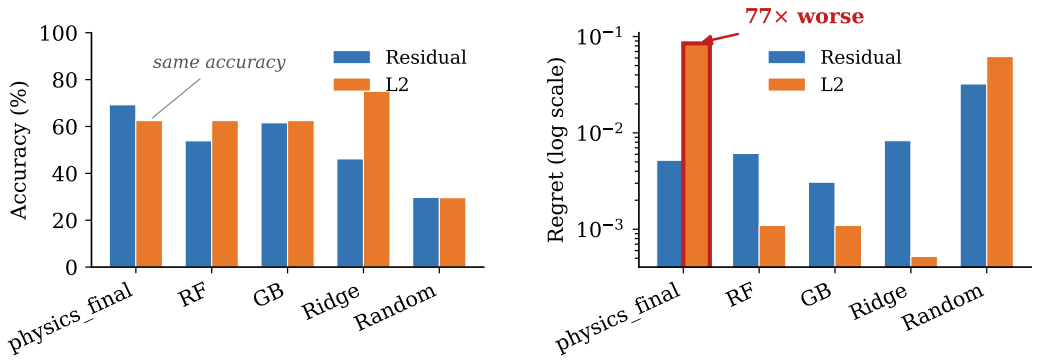
	PDE Residual (13 PDEs)		**Relative L_2 (8 PDEs)**	
Selector	Acc. [95% Wilson CI]	Regret	Acc. (%)	Regret
Base (training-dynamics features only)				
Physics-loss final	69.2% [42.4, 87.3]	0.0052	62	0.0850
Random Forest	53.8% [29.1, 76.8]	0.0061	62	0.0011
Gradient Boosting	61.5% [35.5, 82.3]	**0.0031**	62	0.0011
Ridge	46.2% [23.2, 70.9]	0.0083	**75**	**0.0005**
Training prior	53.8% [29.1, 76.8]	0.0047	50	0.0022
PDE-aware (training + 7 PDE structural features)				
RF + PDE (all)	61.5% [35.5, 82.3]	0.193†	—	—
GBR + PDE (all)	61.5% [35.5, 82.3]	0.205†	—	—
GBR + PDE (top-5, fold-wise)	**84.6**% [57.8, 95.7]	**0.040**†	—	—
RF + PDE (top-5, fold-wise)	**84.6**% [57.8, 95.7]	**0.040**†	—	—
MetaRouter MLP	46.2% [23.2, 70.9]	0.563†	—	—
Random (expected)	29.7%	1.008†	30	0.062

with rel-L_2 defined, 280 samples). **PDE-aware routing** augments either with 7 PDE-characteristic features (Table 7 PDE-aware rows). Best policy varies by PDE and objective (Figure 1; Table 28 in the appendix): `hybrid_full` wins 7/13, `hybrid_no_gate` 4/13, `pinn_only` 2/13 under residual. We evaluate *probe-then-commit* routing: run all K policies for E_p probe epochs, then commit to one for the remaining $E_2 - E_p$ epochs. For K=3, E_p=50, E_2=500 this saves exactly 60% of Stage 2 policy-selection compute, *excluding* Stage 1 data generation, Stage 1 pretraining, and router training. Features: 38-dimensional probe-phase loss statistics (Appendix E); selectors include Random Forest, Gradient Boosting, Ridge regression, a `physics_loss_final` heuristic, training prior, Successive Halving, and LinUCB.

Accuracy and regret, base and PDE-aware. Table 7 combines base routing (training-dynamics features only, LOPO CV) with the PDE-aware extension (seven computable PDE structural features: dimension, nonlinearity/higher-order/reaction flags, derivative order, exact-solution availability, domain volume). Each row reports accuracy *and* regret under both objectives, with Wilson 95% confidence intervals on the 13-PDE residual accuracy. Wilson CIs for base RF 53.8% [29.1, 76.8], GBR 61.5% [35.5, 82.3], and `physics_loss_final` 69.2% [42.4, 87.3] widely overlap, so cross-method rankings should be treated cautiously on this 13-PDE suite. We provide a benchmark-composition bootstrap (Appendix E.1) that preserves the pre-bootstrap ranking but with overlapping percentile CIs. Base rows give *absolute* regret (chosen minus oracle mean metric); the PDE-aware rows and the Random residual entry give *relative* regret, (chosen − oracle)/oracle, on which base RF and the heuristic score 0.208 and 0.069. The base residual regrets and the heuristic and Random Forest L_2 regrets are recomputed from the released fold decisions and runs by `verify_claims.py`. The training prior picks the policy with the lowest mean metric pooled over the held-in PDEs.

Accuracy versus regret, defined. For a held-out PDE, a router *hits* if it picks the oracle-best policy; top-1 accuracy is the fraction of hits. Its *regret* is the mean metric of the policy it picked minus that of the oracle policy (zero on a hit), averaged over held-out PDEs. Accuracy counts misses; regret prices them. A worked case from our L_2 folds: on Wave2D the oracle is `hybrid_full`; the physics-loss heuristic picks `pinn_only`, whose mean relative L_2 is 0.310 higher, so that single miss costs 0.310. Random Forest's three misses (Allen–Cahn, Heat1D, KdV) cost 0.0037, 0.0038 and 0.0013. Both routers are right on 5 of 8 PDEs, yet their mean regrets are 0.0850 and 0.0011.

Accuracy–regret dissociation (the evaluation-methodology finding). Under the L_2 objective, the physics-loss heuristic matches *Random Forest and Gradient Boosting* in top-1 accuracy (62% each); Ridge regression reaches 75% on this objective and is *not* matched by the heuristic. Despite this 62% accuracy tie between the heuristic and Random Forest, their mean regrets are 0.0850 and 0.0011



(a) Routing Accuracy **(b) Routing Regret**

Figure 2: Accuracy–regret dissociation under L_2 routing: the physics-loss heuristic and the Random Forest router have equal top-1 accuracy (62%) but differ by 77× in mean realized regret (0.0850 vs. 0.0011). Ridge regression (75% accuracy on this objective) is not subject to the heuristic-tie comparison.

respectively—a 77× gap between the Random Forest router and the physics-loss heuristic (Figure 2). When the heuristic errs it catastrophically selects a far-from-oracle policy; when the Random Forest router errs it picks a near-oracle one. Two routers with identical top-1 accuracy can differ by nearly two orders of magnitude in regret (excess error over the oracle). **Evaluation-by-accuracy alone is misleading for PDE policy routing; regret must be reported alongside.** Nearly all of the heuristic's L_2 regret comes from two of its three misses, AdvDiff1D (high Péclet, 0.366) and Wave2D (0.310), both choosing `pinn_only`; the third, Heat1D, costs 0.0038. The dissociation appears under L_2 and not under the residual objective, where the heuristic is competitive: physics loss is a direct proxy for residual evaluation but an indirect one for solution accuracy. Restricted to the two 2D PDEs, the three learned routers pick the oracle on both (L_2 regret 0) while the heuristic picks it on one (regret 0.155).

Does the dissociation travel? Rerunning the identical protocol on tabular classifier selection (8 learners, 10 tasks, 10 seeds; no physics, no PINN code) reproduces the dissociation but caps it: across 114 accuracy-tied router pairs the regret ratio has median 1.26× and maximum 3.59×, and the accuracy–regret correlation is -0.805 ± 0.080 rather than -1 (Appendix J). The gap tracks how heavy-tailed best-policy quality is across tasks (largest over smallest best-policy error): 206× on our 8 L_2 PDEs, where the dissociation is 77×; 22× under residual, where the heuristic is competitive; 26× for classifiers, where the gap is at most 3.6×. In these three settings top-1 accuracy was least safe where policy quality was most heavy-tailed; this range can be measured from benchmark runs before building a router.

Leave-one-family-out (LOPO-F) generalization. PDE structural features might let a router recognize a PDE from its siblings rather than learn transferable dynamics, so we retrain every router with the held-out PDE's *whole family* removed from training, still scoring all 13 PDEs. Holding out families costs the PDE-aware routers 15–23 points of PDE-level accuracy and 10–17 points of family-macro accuracy (Table 8): GBR+PDE (top-5) falls from 84.6% to 69.2% and from 89.6% to 79.2% family macro, RF+PDE (top-5) to 61.5% and 72.9%, and their relative regret rises from 0.040 to 0.056 and 0.093. Part of their leave-one-PDE-out accuracy therefore comes from same-family configurations in training. They route no better than the training-free `physics_final` heuristic (69.2%, family macro 81.2%, regret 0.069), and GBR+PDE keeps the lower regret. The base routers barely move, as expected for features without PDE identity (per-family decisions in Table 13).

Caveats. The headline 84.6% needs fold-wise top-5 feature selection (full-feature RF+PDE: 61.5%) and falls to 61.5–69.2% under a family holdout. On a fixed 10/3 split (two test PDEs have siblings in training) RF and the heuristic reach 67% with 16× lower regret than random. The diagnostic regret heuristic (Appendix I) holds on 44/52 PDE–window pairs (84.6%; 11 of 13 PDEs at every $k \in \{10, 20, 50, 100\}$); it is a diagnostic sanity check, not a formal bound. Its two Burgers violations

Table 8: LOPO versus LOPO-F routing (PDE residual, 13 PDEs, 8 families). Family macro: mean over families of the fraction routed to the oracle. `physics_final` trains nothing, so the protocol cannot change it. From `routing_tables.py -table lopof`.

	PDE	PDE-level acc.		Family macro		Fully correct
Selector	features?	LOPO	LOPO-F	LOPO	LOPO-F	(LOPO-F)
Ridge	No	46.2%	46.2%	45.8%	45.8%	2/8
Random Forest	No	53.8%	53.8%	54.2%	56.2%	4/8
Gradient Boosting	No	61.5%	61.5%	68.8%	68.8%	5/8
`physics_final` (heuristic)	No	69.2%	69.2%	81.2%	81.2%	6/8
GBR + PDE (top-5)	Yes	84.6%	69.2%	89.6%	79.2%	5/8
RF + PDE (top-5)	Yes	84.6%	61.5%	89.6%	72.9%	5/8

keep rank transfer ($\rho_k \geq 0.86$); one traces to probe-loss spread, the other is unexplained (Appendix I). Corrections to the submitted version are listed in Appendix L.

9 Discussion, Limitations, and Conclusion

How to use these results.

1. *Score routers by regret as well as top-1 accuracy*, and first measure how heavy-tailed best-policy quality is across your tasks: where it spans two orders of magnitude ($206\times$ here under L_2), accuracy alone misleads (Section 8).
2. *Pretrain, then add a short low-learning-rate stage; causal weights are optional* (uniform weights gain as much, $+6.0\%$ vs. $+5.9\%$); the pretraining advantage held in all 72 noise and sparsity cells of Section 7.
3. *Without pretraining, do not use causal weighting* (worse than uniform weights on 9/9, -3.9%); adaptive ϵ does not recover accuracy.
4. *Distrust the probe on shock problems:* both regret-check violations are Burgers configurations, and no cheap pre-commit test flags both (Appendix I).
5. *Use a hybrid at or near its training regime.* On Wave2D it is 41% better in domain (0.419 vs. 0.706), at parity under 0.8/$1.3\times$ domain rescaling, and twice as bad as `pinn_only` at 1.5–$2\times$ (1.467 vs. 0.726; 5 seeds).

Limitations. (1) Headline results use analytical references; Section 7 stresses them with noise and sparsity but cannot leave that regime, because regret needs an oracle. Real measurement noise, residual-only scoring and inverse problems are untested. (2) The suite is 1D-dominated: 11 of 13 routing configurations are 1D and the robustness probes add only Heat3D and MultiscaleHeat2D; coupled multiphysics PDEs are untested, and on the multi-scale case no policy resolves the fine scale. (3) The policy library is small (3–5 per PDE) on a fixed $200/500/200$ schedule. (4) External methods are positioned, not broadly reproduced; only internal PINO is a same-pipeline baseline; extending the suite costs roughly 0.8–5.8 GPU-hours per equation family (Appendix K). (5) Hybridization is not an OOD-robustness guarantee, the regret check is a diagnostic, not a bound, and late-time weight suppression is not observed at the training ϵ ($w_k \geq 0.94$; Appendix G.3).

Conclusion. *Within this staged-pipeline regime, removing Stage-1 pretraining raises the PDE residual by* 125–694% *at equal training length, causal weighting adds nothing beyond extra training and hurts without pretraining, and a policy router must be evaluated by regret, not accuracy alone, wherever policy quality is heavy-tailed.* In an operator-free probe pipeline, the warm-start part of the pretraining effect survives noise, sparsity and d=3, but on a multi-scale problem it lives entirely in the resolved scale. We release PINNBENCH's code, all run results and a claim checker.

Data, code, and reproducibility. **Code:** `https://github.com/suanlab/PINNBench`. **Artifacts:** `https://huggingface.co/datasets/suanlab/PINNBench` hosts all 3,545 run JSONs and the Croissant file (Appendix M); data: Apache-2.0, code: MIT. `make verify-claims` recomputes the headline numbers (manifest: Appendix L).

Acknowledgments and Disclosure of Funding

This work was supported by the Ministry of Science and ICT and the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (No. RS-2026-25498341 and No. RS-2025-24534935). *Competing interests:* The authors declare no competing interests.

References

Rafael Bischof, Michal Piovarči, Michael A. Kraus, Siddhartha Mishra, and Bernd Bickel. HyPINO: Multi-physics neural operators via HyperPINNs and the method of manufactured solutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 160645–160678, 2025. doi: 10.52202/085713-4845. Spotlight.

Wenqian Chen, Amanda A. Howard, and Panos Stinis. Self-adaptive weights based on balanced residual decay rate for physics-informed neural networks and deep operator networks. *Journal of Computational Physics*, 542:114226, 2025. doi: 10.1016/j.jcp.2025.114226.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conference on Artificial Intelligence*, pages 3460–3468, 2015. URL `https://www.ijcai.org/Abstract/15/487`.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, volume 37, pages 76721–76774, 2024. doi: 10.52202/079017-2442.

Kevin G. Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 240–248, 2016. URL `https://proceedings.mlr.press/v51/jamieson16.html`.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 26548–26560, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/df438e5206f31600e6ae4af72f2725f1-Abstract.html`.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. URL `https://jmlr.org/papers/v18/16-558.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://openreview.net/forum?id=c8P9NQVtmnO`.

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

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.

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

Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning: a survey. *Machine Learning*, 113(11–12):8371–8425, 2024. doi: 10.1007/s10994-024-06619-7.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pages 1596–1611, 2022. doi: 10.52202/068431-0117.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022. doi: 10.1016/j.jcp.2021.110768.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421: 116813, 2024. doi: 10.1016/j.cma.2024.116813.

A Benchmark Specification

All 8 PDE families are implemented in `src/causal_pinn/physics/pde/` and inherit from the abstract `PDEProblem` base class. Table 9 lists equations, parameters, domains, boundary conditions, and exact-solution status.

Table 9: Full PDE definitions with boundary conditions and exact-solution availability, as recorded in each run's `pde_config`. The 8 configurations with an exact solution are the ones routed under relative L_2; the other 5 are scored by PDE residual only.

Config	Domain	Time	Boundary	Exact solution
Heat1D	$x \in [0, 1]$	$[0, 1]$	Dirichlet (zero)	$\sin(\pi x)\, e^{-\pi^2 t}$
AdvDiff1D	$x \in [0, 1]$	$[0, 1]$	Periodic	decaying travelling sine
AdvDiff1D (high Pe)	$x \in [0, 1]$	$[0, 0.5]$	Periodic	decaying travelling sine
Burgers1D	$x \in [0, 1]$	$[0, 1]$	Dirichlet (zero)	none (residual only)
Burgers1D (low ν)	$x \in [0, 1]$	$[0, 1]$	Dirichlet (zero)	none (residual only)
Burgers1D (very low)	$x \in [0, 2]$	$[0, 0.5]$	Dirichlet (zero)	none (residual only)
KdV1D	$x \in [0, 20]$	$[0, 0.5]$	Dirichlet (from exact)	one-soliton (sech^2)
AllenCahn1D	$x \in [0, 2]$	$[0, 1]$	Dirichlet (from exact)	stationary tanh front
AllenCahn1D (sharp)	$x \in [0, 2]$	$[0, 0.5]$	Dirichlet (from exact)	stationary tanh front
Wave2D	$[0, 1]^2$	$[0, 1]$	Dirichlet (zero)	standing mode
ReactDiff1D	$x \in [0, 1]$	$[0, 1]$	Dirichlet (1 / 0)	none (residual only)
ReactDiff1D (stiff)	$x \in [0, 1]$	$[0, 0.5]$	Dirichlet (1 / 0)	none (residual only)
NavierStokes2D	$[0, 2\pi]^2$	$[0, 1]$	Periodic	Taylor–Green

Each PDE provides `residual`, `exact_solution`, `boundary_condition`, and `initial_condition` methods. Adding a new PDE requires implementing these four methods; the training, routing, and statistical pipeline then compose automatically.

B Baseline Protocol Specification

All 5 policies share the same 3-stage schedule; they differ in components activated during Stage 2 and Stage 3 (Table 10).

Table 10: Complete policy specification. All use $200/500/200$ epochs, Adam, LR $10^{-3}/10^{-4}/10^{-5}$, batch 256, 4,000 collocation / 500 boundary points.

Policy	Operator pretrain	Gate	λ_{physics}	Causal (Stage 3)
`pinn_only`	No (PINN backbone only; same Stage 1 supervised data)	—	1.0	Yes
`hybrid_no_gate`	Yes	No	1.0	Yes
`hybrid_full`	Yes	Yes	1.0	Yes
`hybrid_full_lambda0.1`	Yes	Yes	0.1	Yes
`hybrid_full_lambda0.5`	Yes	Yes	0.5	Yes

C Evaluation Metric Definitions

Let $\hat{u}$ be the model prediction and u^* the reference.

Relative L_2 error. $\text{Rel. } L_2 = \|\hat{u} - u^*\|_2 / \|u^*\|_2$.

PDE residual norm. $\|\mathcal{N}[\hat{u}]\|_2$ at evaluation collocation points.

Regret. For held-out PDE d, $\text{Regret}(d) = \mathcal{L}_{\text{eval}}(\hat{\pi}(d), d) - \mathcal{L}_{\text{eval}}(\pi^*(d), d)$, where π^* is the oracle policy on d (mean over 10 seeds).

Top-1 accuracy (routing). Fraction of held-out PDEs where $\hat{\pi} = \pi^*$.

Wilson 95% CI for accuracy. Standard Wilson score interval on the binomial proportion.

BH-FDR. Benjamini–Hochberg step-up procedure on sorted Wilcoxon p-values with q=0.05, applied across all reported tests within a family.

D Full Statistical Tables

All comparisons use exact Wilcoxon signed-rank tests, paired bootstrap 95% CIs, and paired Cohen's d, with Benjamini–Hochberg FDR correction at q=0.05. The full table is in `experiments/analysis/statistical_tests.md` (machine-readable: `statistical_tests.json`); key rows are reproduced below.

Table 11: Core statistical support for the main claims (10 seeds unless noted). Negative mean difference favors the first-listed method. The 4 in-domain hybrid-vs-`pinn_only` rows are significant after BH-FDR, as is the Wave2D OOD row (10 seeds, one pair per seed with the two shifted speeds averaged); the 2 PINO comparisons are also significant; 3/5 causal-loss rows are significant. Causal rows are with- minus without-causal-loss runs of the original single-PDE studies (not the 2×2 ablation of Table 5); there a positive difference means causal loss raised the residual, so the significant Burgers1D row is a degradation.

Comparison	Metric	Mean diff.	95% CI	Wilcoxon p	Cohen's d
Heat1D hybrid vs PINN	Rel. L_2	-0.378	$[-0.404, -0.351]$	0.002	-8.33
Wave2D hybrid vs PINN	Rel. L_2	-0.274	$[-0.317, -0.227]$	0.002	-3.55
KdV1D hybrid vs PINN	Rel. L_2	-0.083	$[-0.222, -0.008]$	0.004	-0.38
AllenCahn1D hybrid vs PINN	Rel. L_2	-0.201	$[-0.245, -0.173]$	0.002	-3.12
Wave2D OOD hybrid vs PINN	Rel. L_2	-0.061	$[-0.075, -0.047]$	0.002	-2.58
Hybrid vs PINO Heat1D	Rel. L_2	-0.745	$[-0.755, -0.734]$	0.002	-41.28
Hybrid vs PINO AdvDiff1D	Rel. L_2	-0.017	$[-0.024, -0.010]$	0.004	-1.40
Heat1D causal (5s)	Rel. L_2	0.002	$[-0.001, 0.005]$	0.312	0.53
Burgers1D causal	PDE residual	0.015	$[0.011, 0.020]$	0.002	1.97
AdvDiff1D causal (5s)	Rel. L_2	-0.015	$[-0.018, -0.010]$	0.063	-2.81
KdV1D causal	PDE residual	-0.001	$[-0.001, -0.001]$	0.002	-2.52
AllenCahn1D causal	PDE residual	-0.002	$[-0.004, -0.001]$	0.027	-0.80

E Per-PDE Routing Decisions and Family Holdout

Feature list (38-dimensional). For each probe-phase loss stream (physics, data, BC, total): mean, std, min, initial, final, slope ($6 \times 4 = 24$ features). Cross-objective ratios at final probe epoch: physics/data, physics/total, BC/total (3 features). Per-loss decay ratios (final/initial): physics, data, BC, total (4 features). Physics-weight (λ_{physics}) statistics: initial, final, mean, std, min, slope, final/initial ratio (7 features; constant within a policy, so they carry zero importance). Total: 38.

PDE structural features (7). Spatial dimension, nonlinear-term flag, higher-order-term flag, reaction-term flag, derivative order, exact-solution availability, domain volume.

Feature importance. Top-5 features by RF importance in the earlier 8-group early-predictor run (`rq_early_predictor_expanded`, E_p=50). Residual objective: `physics_loss_final` 37.8%, `physics_loss_mean` 9.9%, `physics_loss_min` 7.6%, `physics_loss_std` 7.0%, `total_loss_initial` 5.1% (cumulative 67.4%). L_2 objective (`rq_early_predictor_rel_l2`) spreads importance across cross-objective ratios, none above 10%. On the 13-PDE suite the ranking is more concentrated: `physics_loss_min` (35–36%) and `physics_loss_final` (33%) lead under residual.

Scaling analysis. 5 PDEs: RF 80.0%, physics-loss 80.0%, random 27.3%. 9 PDEs: 77.8%, 77.8%, 28.1%. 13 PDEs: 53.8%, 69.2%, 29.7%. Routing difficulty grows with PDE diversity.

Table 12: Per-PDE routing decisions (PDE residual, $E_p = 50$, leave-one-PDE-out): the oracle policy and whether each selector picked it. Generated by `routing_tables.py -table perpde` from `experiments/results/paper_combined/routing_evaluation/results.json`.

Held-out PDE	Oracle	RF	GB	Ridge	`physics_final`
Heat1D	`hybrid_full`	✓	✓	×	✓
AdvDiff1D	`hybrid_no_gate`	✓	✓	×	✓
AdvDiff1D (high Pe)	`pinn_only`	×	✓	×	✓
Burgers1D	`hybrid_full`	×	×	×	×
Burgers1D (low ν)	`hybrid_no_gate`	✓	×	✓	×
Burgers1D (very low)	`hybrid_no_gate`	×	×	✓	×
KdV1D	`hybrid_no_gate`	×	×	✓	✓
AllenCahn1D	`hybrid_full`	✓	✓	×	✓
AllenCahn1D (sharp)	`hybrid_full`	✓	✓	✓	✓
Wave2D	`pinn_only`	×	✓	×	✓
ReactDiff1D	`hybrid_full`	✓	✓	✓	✓
ReactDiff1D (stiff)	`hybrid_full`	×	×	×	×
NavierStokes2D	`hybrid_full`	✓	✓	✓	✓
Correct	—	7/13	8/13	6/13	9/13

Leave-one-family-out (LOPO-F) full decisions. Table 13 gives, for each family, the fraction of its configurations each base selector routes to the oracle when the whole family is held out of training.

Table 13: Family-level holdout decisions for the base selectors (PDE residual, $E_p = 50$): ✓ all configurations of the family correct, × none, otherwise the fraction. Family macro is the mean of these fractions. Generated by `routing_tables.py -table family` from `experiments/results/paper_combined/routing_family_holdout/results.json`.

Family	n	Oracle polic(ies)	RF	GB	Ridge	`physics_final`
Heat	1	`hybrid_full`	✓	✓	×	✓
AdvDiff	2	`hybrid_no_gate`, `pinn_only`	✓	✓	×	✓
Burgers	3	`hybrid_full`, `hybrid_no_gate`	×	×	2/3	×
KdV	1	`hybrid_no_gate`	×	×	✓	✓
AllenCahn	2	`hybrid_full`	✓	✓	1/2	✓
Wave	1	`pinn_only`	×	✓	×	✓
ReactDiff	2	`hybrid_full`	1/2	1/2	1/2	1/2
NavierStokes	1	`hybrid_full`	✓	✓	✓	✓
Family macro			56.2%	68.8%	45.8%	81.2%
Fully correct			4/8	5/8	2/8	6/8

Budget savings: what 60% does and does not include. $B_{\text{route}} = K \cdot E_p + (E_2 - E_p) = E_2 + (K-1)E_p$. For K=3, E_p=50, E_2=500: $B_{\text{route}} = 600$ vs. full-grid $K \cdot E_2 = 1500$, i.e. 60% Stage 2 savings. This is a Stage-2 policy-selection-only quantity. It does *not* include: (i) Stage-1 operator pretraining cost, (ii) reference-data generation cost, (iii) Stage-3 causal calibration cost, or (iv) inference cost. Practitioners reporting end-to-end training-system efficiency should add these terms separately.

Held-out 3-PDE generalization. Train on 10 PDEs, evaluate on 3 unseen (AllenCahn sharp, Burgers very-low-ν, Navier–Stokes 2D): 67% accuracy, mean regret 0.002 (16× lower than random). L_2 routing on 2 held-out PDEs: 100% accuracy, zero regret. This is a small held-out sanity check; it does not establish generalization to arbitrary PDE families (the LOPO-F result above is the stronger family-level test).

E.1 Benchmark-composition bootstrap stability

The bootstrap preserves the pre-bootstrap ranking but the percentile CIs overlap: at the 13-PDE benchmark size, point estimates of routing accuracy should be read as ranges, and cross-method ranking claims ("X% better than Y") are unstable to which 9-of-13 PDEs are drawn.

Table 14: Routing accuracy under bootstrap resampling of the 13-PDE suite (sample 9, replacement, 1000 reps, seed 42; PDE-residual objective). The pre-bootstrap ranking (`physics_loss_final` > GBR > RF > Ridge) is preserved across bootstrap means; percentile CIs overlap, so cross-method point-rank claims at the 13-PDE benchmark size are unstable. Source: `experiments/scripts/paper_a_routing/bootstrap_stability.py`.

Selector	Mean (%)	SD (%)	95% pct. CI
`physics_loss_final`	69.1	15.2	[33.3, 100.0]
Gradient Boosting	62.2	15.5	[33.3, 88.9]
Random Forest	53.3	16.2	[22.2, 88.9]
Ridge	46.9	16.3	[11.1, 77.8]

F External Baseline Probe Details

Table 15: HyPINO official native-suite smoke evaluation, rerun by us from the released checkpoint on the upstream benchmark files. Five of seven benchmarks reproduce the published HyPINO values (MSE/SMAPE, Bischof et al., 2025, Table 1) to their reported precision and `poisson_G` nearly does (SMAPE 60.5% vs. 61%); `helmholtz_G` does not (0.378 / 112.8% here against 0.13 / 64% published), and we have not isolated the cause. Reported for executability and context only; not directly comparable to our staged exact-solution suite.

Benchmark	MSE	MAE	Max error	SMAPE
heat	2.315×10^{-2}	1.353×10^{-1}	2.982×10^{-1}	42.07%
helmholtz	5.692×10^{-3}	5.983×10^{-2}	2.257×10^{-1}	36.04%
helmholtz_G	3.781×10^{-1}	5.162×10^{-1}	1.028	112.79%
poisson_C	5.550×10^{-2}	1.915×10^{-1}	5.146×10^{-1}	86.17%
poisson_L	1.754×10^{-4}	9.319×10^{-3}	6.716×10^{-2}	39.25%
poisson_G	1.767×10^{-1}	3.456×10^{-1}	7.774×10^{-1}	60.47%
wave	2.853×10^{-1}	4.188×10^{-1}	1.481	149.52%

Table 16: HyPINO adapter and same-task fine-tuning probes. Heat1D and AdvectionDiffusion1D are mapped to HyPINO coordinates $(X, Y) \in [-1, 1]^2$ via $X = 2x - 1$, $Y = 2t - 1$. Fine-tuning uses 500 Adam steps, 3 seeds. Adaptation probes only—not a full HyPINO retraining. Our hybrid is trained from scratch over $200/500/200$ epochs and is *not* budget-comparable with HyPINO's foundation-checkpoint adaptation; we do not claim to beat HyPINO on Heat1D.

Probe	PDE	Rel. L_2	MSE	Note
Zero-shot adapter	Heat1D	2.412	0.153	Compatibility only
Target PINN + 500-step adapt.	Heat1D	0.0115 ± 0.0005	$(3.48 \pm 0.33)\times10^{-6}$	3 seeds
Target PINN + 500-step adapt.	AdvDiff1D	0.0043 ± 0.0001	$(1.20 \pm 0.05)\times10^{-6}$	3 seeds
`pinn_only` (ours, 10 seeds)	Heat1D / AdvDiff1D	$0.397/0.587$	—	Context
Hybrid full (ours, 10 seeds)	Heat1D / AdvDiff1D	$0.018/0.018$	—	Context

Table 17: PINNacle executable subset smoke suite (200 iterations, 2×32 hidden units; validates protocol compatibility, not PINNacle reproduction or method ranking).

Task	Iter.	Rel. L_2	MSE	MAE	Runtime
Burgers1D	200	0.633	0.149	0.293	21.1s
Wave1D	200	1.142	0.407	0.519	80.9s
Poisson1D	200	0.950	0.452	0.518	96.9s
Helmholtz2D	200	1.713	0.766	0.710	134.8s
Heat2D multiscale	200	1.014	0.035	0.103	153.7s
Mean $\pm$ std.	200	1.090 ± 0.395	0.362 ± 0.285	0.429 ± 0.234	97.5 ± 51.6s

F.1 HyPINO matched-budget comparison

Table 18 gives the hybrid and HyPINO the same wall-clock budget on four PDEs.

Table 18: HyPINO matched-wall-budget comparison (10 seeds $\times$ 4 PDEs). The hybrid trained from-scratch under HyPINO's measured per-call wall budget loses to HyPINO finetuned from a foundation checkpoint, by $20\times$ on Heat1D and $90\times$ on AdvDiff1D. This is consistent with, but does not isolate, the role of pretraining: HyPINO differs from our hybrid in architecture and pretraining data as well as in starting point, and at this wall budget from-scratch training is insufficient. This experiment does not invalidate the hybrid pipeline, but it sharpens the conditions under which from-scratch hybrid training is competitive (with full multi-stage budget; not under matched-fine-tune budget). Burgers1D has no exact solution in this run; a rerun against an exact Burgers solution gives 0.355 ± 0.133 (PINN fallback) vs. 0.354 ± 0.026 (hybrid), 10 seeds. On Burgers1D and AllenCahn1D the HyPINO branch falls back to PINN-only fine-tuning because neither is in HyPINO's operator vocabulary. Source: `experiments/results/paper_combined/hypino_matched_budget/aggregate.json`.

PDE	HyPINO finetune / PINN-fallback (rel. L_2)	Hybrid matched (rel. L_2)	n
Heat1D	0.0108 ± 0.0006 (HyPINO finetune)	0.2163 ± 0.0502	10
AdvDiff1D	0.0047 ± 0.0008 (HyPINO finetune)	0.4277 ± 0.1552	10
AllenCahn1D	0.1681 ± 0.0255 (PINN fallback)	0.3506 ± 0.0283	10
Burgers1D	NaN (no exact solution in this run)	NaN	10

Table 19: Wang-style single-stage causal comparison, 10 seeds. Single-stage causal gains are small and metric-dependent; the full staged pipeline dominates under both metrics.

PDE	Single, no causal	Single, causal	Full staged
AdvDiff1D (L_2)	0.878	0.869	0.018
Heat1D (L_2)	1.028	1.012	0.020
KdV1D (L_2)	0.370	0.349	0.006
AllenCahn sharp (L_2)	0.533	0.532	0.077
AdvDiff1D residual	0.0469	0.0443	0.0161
Heat1D residual	0.0470	0.0476	0.0104
KdV1D residual	0.1228	0.1273	0.0134
AllenCahn sharp residual	0.2344	0.2366	0.0459

G Causal Mechanism Details

Gradient alignment. For each of Heat1D, AdvDiff1D, KdV1D and sharp AllenCahn1D (seed 42), we compute $\cos(\nabla_\theta \mathcal{L}_{\text{causal}}, \nabla_\theta \mathcal{L}_{\text{PDE}})$ every 10 epochs over Stages 2 and 3. Means are at least 0.999 and minima at least 0.997 across all four PDEs, ruling out gradient conflict.

G.1 Gradient cosine summary across 4 PDEs

The $\cos \approx 1.0$ finding generalizes across all 4 analyzed PDEs: the causal and physics gradients are parallel in every (PDE, condition) pair, which rules out directional conflict between $\nabla_\theta \mathcal{L}_{\text{causal}}$ and

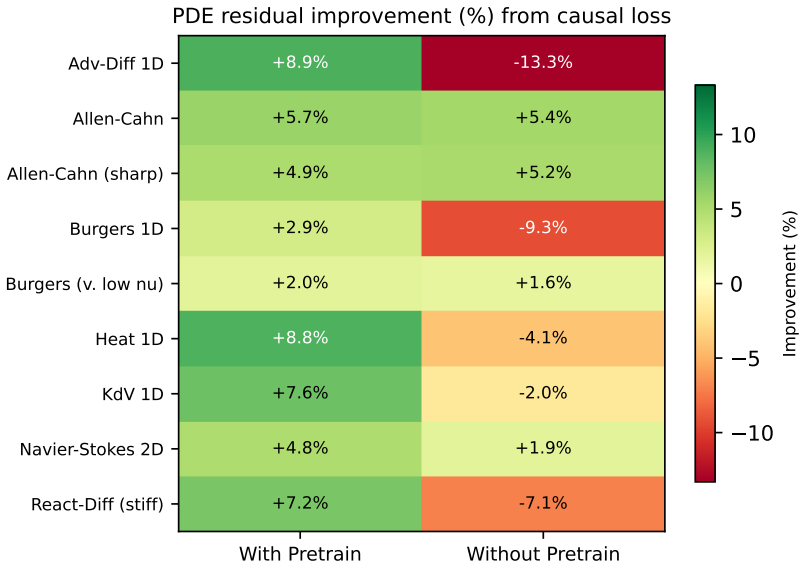


Figure 3: 2×2 stage ablation heatmap: color encodes PDE-residual change $\Delta\%$ relative to the No-Causal baseline within each pretraining condition. The causal-calibration stage helps with pretraining on all 9 PDEs (left block), but a uniform-weight stage of the same length does as well (Table 21); without it (right block) it raises the residual on 5 of 9 (3 significant).

Table 20: Gradient cosine similarity between physics and causal directions across 4 PDEs and 2 pretraining conditions. The $\cos \approx 1.0$ finding from Heat1D (mechanism discussion in Section 6) generalizes: every (PDE, condition) pair shows mean cosine ≥ 0.9993 and minimum ≥ 0.9974. Redundancy, not directional conflict, is the consistent gradient-flow phenomenon. Source: `experiments/scripts/paper_c_causal/gradient_cosine_summary.py`.

PDE	Condition	cos min	cos max	cos mean
Heat1D	with_pretrain	0.9997	1.0000	1.0000
Heat1D	without_pretrain	1.0000	1.0000	1.0000
AdvDiff1D	with_pretrain	0.9974	1.0000	0.9993
AdvDiff1D	without_pretrain	1.0000	1.0000	1.0000
KdV1D	with_pretrain	1.0000	1.0000	1.0000
KdV1D	without_pretrain	1.0000	1.0000	1.0000
AllenCahn1D (sharp)	with_pretrain	0.9999	1.0000	1.0000
AllenCahn1D (sharp)	without_pretrain	0.9999	1.0000	1.0000

$\nabla_\theta \mathcal{L}_{\text{PDE}}$ on these runs; gradient magnitudes are not reported here. This addresses the concern that the "redundancy not conflict" claim is Heat1D-specific.

Weight dynamics. On Heat1D (seed 42), with weights evaluated at a diagnostic ϵ=100: *with pretraining* the weights of bins 2–100 rise from 0.33 to 0.96 (mean 0.89) and barely change over Stage 3; *without pretraining* they sit at 0.14–0.46 (mean 0.29). Large early residuals thus suppress later-time weight, but not to zero, and at the training $\epsilon \leq 1$ the suppression is mild (Appendix G.3).

Temporal error distribution. Late-time/early-time relative L_2 ratios (mean of the last three over the first three time slices, full pipeline without causal loss, seed 42): Heat1D $672\times$, AdvDiff1D $20.7\times$, KdV1D $1.16\times$, sharp AllenCahn1D $0.98\times$. The two high-ratio PDEs gain the most from the causal-calibration stage with pretraining ($+8.8\%$, $+8.9\%$), but uniform weights gain as much (Table 21), so the ratio does not describe a reweighting effect; the near-unity AllenCahn ratio suggests its benefit comes from interface ordering rather than temporal reweighting.

Pretraining-effect magnitudes (PDE residual, Stage 1 removed). Removing Stage 1 raises the residual (no causal loss) by 122–933% across the 9 ablated PDEs: Burgers1D 122%, sharp AllenCahn1D 300%, Burgers1D (ν=10^{-4}) 387%, NavierStokes2D 389%, stiff ReactDiff1D 447%, AllenCahn1D 614%, Heat1D 781%, KdV1D 875%, AdvDiff1D 933%. This dwarfs the effect of the causal-calibration stage ($+2$ to $+9$% with pretraining, -13 to $+5$% without). Source: `experiments/results/paper_c/ablation_statistics/results.json`.

Epoch-matched controls. Table 21 gives the per-PDE results of the controls summarised in Section 6. Removing Stage 1 with total epochs matched still raises the residual by 125–694%, against 122–933% unmatched.

Table 21: Epoch-matched controls for the 2×2 ablation (PDE residual, 10 seeds, Δ% = (reference − treatment)/reference, so positive favours the treatment; * BH-FDR significant over the 54 tests). Reweighting: causal vs. uniform-weight Stage 3 (same epochs). Extra training: uniform-weight Stage 3 vs. none. Pretraining: Stage 1 + 500 Stage-2 epochs vs. no Stage 1 + 700; the Stage-2 physics-weight ramp scales with its length, so this arm also ramps more slowly (as a reduction relative to the unpretrained arm; the 125–694% in the text is the same effect relative to the pretrained arm). Generated by `control_statistics.py -latex` from `experiments/results/paper_c/control_statistics/results.json`.

	Reweighting		Extra training		Pretraining
PDE	w/ pretrain	w/o	w/ pretrain	w/o	(matched)
Heat1D	-0.0^*	-2.4^*	$+8.8^*$	-1.6	$+84.4^*$
AdvDiff1D	-0.1^*	-14.3^*	$+9.0^*$	$+0.8$	$+84.5^*$
Burgers1D (ν=0.01)	-1.2^*	-7.6^*	$+4.0$	-1.5	$+55.6^*$
Burgers1D (ν=10^{-4})	-0.0^*	-1.3^*	$+2.0$	$+2.9^*$	$+73.9^*$
KdV1D	$+0.0$	-0.8^*	$+7.6^*$	-1.2	$+87.4^*$
AllenCahn1D (ε=0.01)	-0.0^*	-0.5^*	$+5.7$	$+5.8^*$	$+76.4^*$
AllenCahn1D (ε=0.001)	-0.0^*	-0.4^*	$+4.9^*$	$+5.6^*$	$+60.6^*$
ReactDiff1D (stiff)	-0.1^*	-5.8^*	$+7.2^*$	-1.2	$+71.1^*$
NavierStokes2D	-0.0^*	-2.1^*	$+4.9^*$	$+3.9^*$	$+68.3^*$
Mean	-0.2	-3.9	$+6.0$	$+1.5$	$+73.6$

G.2 Pretraining-budget sweep

Table 22 varies the Stage-1 budget and compares causal calibration with no Stage 3 at each budget.

G.3 Clamping ablation null result

Table 23 reports the minimum causal weight with and without a clamping floor of 0.1.

Table 22: Pretraining-budget sweep (10 seeds $\times$ 3 PDEs $\times$ 4 Stage-1 budgets $\times$ 2 causal conditions $= 240$ runs). Causal benefit is small ($\leq 5.3\%$) and not monotonic in pretraining budget. The sweep scores relative L_2, whereas Table 5 scores PDE residual, so its values are not comparable with the $+5.9\%$ there; at budgets below 400 the differences are within seed noise. The sweep compares causal with no Stage 3, so like Table 5 it does not isolate reweighting (Table 21); the dose-response curve is not clean. Source: `experiments/results/paper_combined/pretrain_budget_sweep/aggregate.json`.

PDE	Budget	No-causal rel-L_2	Causal rel-L_2	$\Delta\%$
Heat1D	50	0.943 ± 0.067	0.931 ± 0.038	$+1.3\%$
Heat1D	100	0.648 ± 0.067	0.626 ± 0.066	$+3.4\%$
Heat1D	200	0.206 ± 0.054	0.198 ± 0.045	$+4.1\%$
Heat1D	400	0.0473 ± 0.0170	0.0478 ± 0.0186	-1.0%
AdvDiff1D	50	0.856 ± 0.015	0.845 ± 0.017	$+1.3\%$
AdvDiff1D	100	0.790 ± 0.029	0.783 ± 0.031	$+0.9\%$
AdvDiff1D	200	0.404 ± 0.148	0.399 ± 0.156	$+1.2\%$
AdvDiff1D	400	0.0420 ± 0.0155	0.0397 ± 0.0165	$+5.3\%$
AllenCahn1D	50	0.472 ± 0.025	0.467 ± 0.024	$+1.0\%$
AllenCahn1D	100	0.383 ± 0.042	0.373 ± 0.045	$+2.5\%$
AllenCahn1D	200	0.315 ± 0.027	0.318 ± 0.022	-1.1%
AllenCahn1D	400	0.187 ± 0.031	0.180 ± 0.032	$+4.1\%$

Table 23: Clamping ablation null result. Across 9 PDEs $\times$ 10 seeds in the no-pretraining causal condition, the seed-mean final minimum causal weight ranged from 0.942 (AllenCahn) to 0.997 (Heat1D), and no weight along any trajectory fell below 0.906—far above the proposed clamping floor of 0.1. Clamping therefore left the PDE residual unchanged on all 9 PDEs and rel-L_2 unchanged on the 5 with an exact solution (identical to 4 decimal places). Causal-weight collapse, the mechanism discussed in Section 6, is therefore not observed at this training horizon. We hypothesize that longer Stage-3 training (> 200 epochs) or larger ϵ would trigger collapse; we leave this as future work. Source: `experiments/results/paper_combined/clamping_ablation/aggregate.json`.

PDE	Seed-mean `final_min_weight` (unclamped $\equiv$ clamped)
Heat1D	0.997
AdvDiff1D	0.985
Burgers1D	0.965
Burgers1D (very low ν)	0.977
AllenCahn1D	0.942
AllenCahn1D (sharp)	0.945
KdV1D	0.984
ReactDiff1D	0.985
ReactDiff1D (stiff)	0.954

H Adaptive-ϵ Sensitivity

With pretraining, adaptive ϵ raises the minimum weight modestly (AdvDiff1D $0.780 \to 0.873$) and improves relative L_2 by 2–3% (Heat1D, AdvDiff1D, Allen–Cahn). Without pretraining it keeps $\min w \geq 0.96$, but relative L_2 stays near failure on Heat1D (1.09) and AdvDiff1D (0.96) and reaches only 0.53 on AllenCahn (Burgers has no exact solution), and since no fixed-ϵ run without pretraining exists we cannot claim it prevents a collapse. Adaptive ϵ is a safe setting, not a substitute for pretraining.

I Regret Heuristic: Template, Assumptions, and Failure Modes

Template form. Probe-then-commit trains all P candidate policies for k epochs, ranks them by probe loss, and commits the winner to the full budget. The diagnostic compares observed regret with

$$\mathrm{Regret}(k) \;\leq\; \frac{2\sigma_k\sqrt{2\ln(2P/\delta)}}{\rho_k},$$

Table 24: Minimum causal weight at training end under fixed and adaptive ϵ (higher is less collapsed), mean over 5 seeds. No fixed-ϵ run without pretraining was made, hence the dashes. AllenCahn is the sharp configuration (ε=0.001). Source: `experiments/results/paper_combined/adaptive_epsilon/results.json`.

PDE	Condition	Min w (fixed)	Min w (adaptive)
Heat1D	with pretrain	0.976	0.972
AdvDiff1D	with pretrain	0.780	0.873
Burgers1D	with pretrain	0.872	0.900
AllenCahn	with pretrain	0.959	0.969
Heat1D	no pretrain	—	0.997
AdvDiff1D	no pretrain	—	0.985
Burgers1D	no pretrain	—	0.969
AllenCahn	no pretrain	—	0.964

read as holding with probability at least $1 - \delta$ (δ=0.05 in every check). The probe loss is the mean Stage-2 total training loss over the first k epochs; σ_k is its across-seed standard deviation averaged over policies; ρ_k is the Spearman correlation, over all policy×seed runs, between the probe loss and the final Stage-2 total training loss. Observed regret is measured in the evaluation metric, so the comparison mixes scales and is a heuristic. It rests on four assumptions: (A1) *rank transfer*, $\rho_k > 0$; (A2) *spread calibration*, probe-time σ_k tracks the dispersion of final losses; (A3) sub-Gaussian probe estimates; (A4) a fixed policy set. It omits a finite-sample proof and is a diagnostic, not a theorem.

Empirical check. Observed regret is that of the heuristic the template bounds: rank the policies by total training loss at the end of the k-epoch probe, commit to the lowest, and score the pick by PDE residual against the oracle, leave-one-PDE-out. Over 13 PDEs × $k \in \{10, 20, 50, 100\}$ the check holds on 44/52 pairs (84.6%), 11 of 13 at every window (mean ρ_k 0.826, 0.836, 0.856, 0.874). Where it holds with nonzero observed regret (28 of 44 pairs) the bound exceeds observed regret by a median factor of 6.4 (mean 31.0, pulled up by one outlier at 113). Source: `experiments/results/paper_combined/regret_bound_v13/regret_bound_validation.json`, regenerated by `validate_regret_bound.py`.

k=20 is evaluated in this version. The submitted validator read observed regret from a routing file that was never released, keyed by probe window; it found nothing at k=20 and counted those rows as violations, which produced the "universal failure at k=20" of our submission and rebuttal (withdrawn). Its values for k=10, 50, 100 were one k=50 value per PDE reused at every window. The validator now recomputes observed regret at each window from the released runs; at k=10 this changes two PDEs (AdvDiff1D 0.024 → 0, KdV1D 0 → 0.005) and no verdict.

The Burgers failures are not rank-transfer failures. The two violating configurations, `burgers1d` and `burgers1d_lownu`, have ρ_k between 0.86 and 0.96, close to the suite mean (0.83–0.87), so rank transfer (A1) holds. KdV1D, with the weakest ranking (ρ=0.17 at k=10, 0.49 at k=100), satisfies the check at every window, its regret being 0 or 0.005. For `burgers1d_lownu` the failure is one of spread calibration (A2): at k=50 the probe spread is σ=0.0033 while the gap between the best two policies is 0.061 (19×), so probe losses look alike while final outcomes diverge. For `burgers1d` the two are comparable (0.9×), as they are for AdvDiff1D high-Pe (0.95×), which passes, so spread does not explain that failure and comparing σ_k with the best-to-second gap is not a reliable pre-commit test. This corrects our rebuttal, which attributed both failures to shocks breaking smooth early dynamics.

J Dissociation Transfer Details

Tabular classifier selection: 8 learners over 10 tasks (4 scikit-learn built-ins, 6 synthetic), leave-one-task-out routing, 10 seeds (the seed controls task generation, splits and learner randomness). The driver imports nothing from the PINN code. Six of the eleven routers are shown; all are in `dissociation_transfer/results.json`.

Table 25: Classifier selection, mean ± std over 10 seeds (source: `experiments/results/paper_combined/dissociation_transfer/results.json`). The probe heuristic is within noise of the always-random-forest policy on accuracy but carries about twice its regret.

Router	top-1 accuracy	mean regret
`always_rbf_svm`	$39.0\% \pm 10.4$	0.0239 ± 0.0102
`learned_router`	$35.0\% \pm 11.2$	$\mathbf{0.0187 \pm 0.0075}$
`always_random_forest`	$28.0\% \pm 9.8$	0.0210 ± 0.0048
`probe_heuristic`	$25.0\% \pm 8.1$	0.0413 ± 0.0110
`always_logreg`	$15.0\% \pm 9.2$	0.0706 ± 0.0143
`always_knn`	$8.0\% \pm 7.5$	0.0528 ± 0.0082

Dynamic range is the largest over the smallest best-policy error across tasks, computed identically for both domains: median 25.7 over seeds for classifier selection, 206 for the 8 PDEs routed under L_2, and 22 under the residual objective.

K Compute and Environment

Cost of extending the benchmark. A new equation family needs a `PDEProblem` subclass and 30–50 runs (3–5 policies × 10 seeds). Recorded wall-clock for the full $200/500/200$ pipeline on one TITAN Xp is a median of 73–475 s per run depending on the PDE (35–739 s overall; runs shared GPUs), and the recorded totals are 0.8–5.8 GPU-hours per family. Our rebuttal quoted 6–51 s, which were PINN-only and reduced-budget runs.

Hardware. 4×NVIDIA TITAN Xp (12 GB VRAM each), CUDA 12.1, PyTorch 2.5.1+cu121, Python 3.12.

Runtime. Per-run times are given above. The recorded training time summed over every released run, including reruns and probes, is about 107 GPU-hours. Routing evaluation (feature extraction and cross-validation) runs on CPU.

Software. Repository is a Python package (`causal_pinn`). Dependencies: PyTorch 2.5.1, scikit-learn, NumPy, SciPy. `pyproject.toml` and `requirements-lock.txt` pin versions.

L Reproducibility Manifest

Corrections relative to the submitted version. The following statements of the submitted paper or of the public discussion period are corrected in this version. (i) Stage 2 also carries the supervised data term for every policy whenever an exact solution exists; the submitted version described the reference data only as a Stage-1 channel. (ii) The regret check's "universal failure at k=20" was an artifact of a validator that counted never-evaluated rows as violations; the claim is withdrawn, and k=20 is now evaluated (11/13, as at the other windows). (iii) The rebuttal attributed both Burgers violations to shocks breaking smooth early dynamics; rank transfer is intact on both (Appendix I). (iv) The rebuttal said the hybrid never ends up worse than `pinn_only` out of domain; it is about twice as bad at 1.5–2× domain rescaling. (v) The rebuttal's per-run costs of 6–51 s were PINN-only and reduced-budget runs (Appendix K). (vi) Without pretraining, the causal-calibration effect is -2.4%, not -4.2%, and 6/9 (not 7/9) PDEs improve significantly with pretraining. The submitted paper credited that stage's gain to causal reweighting; epoch-matched controls (Table 21) show it is extra training. (vii) The submitted "leave-one-family-out" numbers regrouped leave-one-PDE-out folds; Table 8 now retrains without the held-out family. (viii) The submitted run count of 1,539 cannot be reproduced from the artifacts and is replaced by the count in Appendix L. (ix) The submitted Croissant appendix described `recordSets` that the released file does not contain, and contained a placeholder instead of a validation report.

Run count. The 3,545 recorded runs are 450 in the 13-PDE routing suite, 360 in the 2×2 stage ablation, 459 single-PDE studies (Section 5), 1,066 causal and diagnostic runs (including 360 epoch-matched controls), and 1,210 robustness probes; a run is any record with a training time and seed,

counted once across copies by `verify_claims.py`. The submitted version quoted 1,539 runs from a tally that the artifacts do not reproduce (about 1,540 runs predate submission by recording date); the count above replaces it.

Table 26: Hyperparameters of the 13-PDE suite and the stage ablation. Scripts hardcode their values; experiments that differ (the single-PDE studies of Section 5, the clamping and adaptive-ϵ studies with 400 Stage-2 epochs and 2,000/200 points, the gradient-cosine and causal-weight diagnostics with 300 Stage-2 epochs and 2,000/200 points, the operator-free robustness probes) are listed here or say so where they are reported. `experiments/configs/default.yaml` documents this protocol and is not read at runtime.

Component	Setting
PINN backbone	
Hidden layers / width	4×64, GELU
FNO operator	
Fourier modes / width / layers	8 / 32 / 4
Causal gate	
Hidden dim / output	32 / Sigmoid
Training	
Optimizer	Adam
Stage 1 LR / Epochs	10^{-3} / 200
Stage 2 LR / Epochs	10^{-4} / 500
Stage 3 LR / Epochs	10^{-5} / 200
Batch size / colloc. / bd.	256 / 4,000 / 500
Causal calibration	
ϵ / time bins	$0.01 \rightarrow 1$ (linear) / 100
Evaluation	
Colloc. / bd. / init.	2,000 / 200 / 200
Seeds (10-seed expts.)	$\{42, 123, 456, 789, 1024, 2048, 3072, 4096, 5120, 6144\}$
Hardware	4×TITAN Xp (12 GB), CUDA 12.1

Extending the benchmark. Adding a new PDE requires implementing the `PDEProblem` interface (residual, exact solution, boundary and initial conditions). Adding a new training policy requires defining stage-specific loss configurations. Adding a new routing method requires implementing a `predict(features)` $\rightarrow$ `policy` interface and running the LOPO evaluation script. The modular design ensures that extensions compose with existing infrastructure.

Best-policy-per-PDE for reference. Table 28 summarizes oracle policies per PDE under the residual objective.

M Croissant Metadata Specification

The released dataset repository (`https://huggingface.co/datasets/suanlab/PINNBench`) carries a Croissant 1.1 file, `croissant.json`, whose core metadata and `recordSets` are generated by Hugging Face from the repository's tabular configurations, with Responsible-AI fields added by us. We describe the file as released (Appendix L lists the change from the submitted version).

Core fields. `name` (`pinnbench`), `version` (`1.3.0`), `license` (Apache-2.0), `url`, `citeAs`, `datePublished`, and a `distribution` listing the repository and one Parquet file set per configuration. There are three configurations, each exposed as a `recordSet` with a matching split set: `adaptive_epsilon`, `meta_router`, and `regret_bound_validation`. The full 3,545-run JSON tree is distributed as repository files and indexed by the manifest (Table 27) rather than by per-record Croissant entries.

RAI fields. Thirteen `rai:` fields are populated: `dataCollection`, `dataPreprocessing`, `dataAnnotationProtocol`, `dataLimitations`, `dataBiases`, `dataUseCases`, `dataSocialImpact`, `hasSyntheticData` (`true`), `personalSensitiveInformation`, `deidentificationMethod`, `safetyMeasures`, `fairness`, and `releaseMaintenance`. They state that every record comes from controlled runs on synthetic PDE problems with analytical

references, that no human annotation or personal data is involved, and that the suite is dominated by 1D problems.

Representative `recordSet` (abridged from the released file).

```
recordSet: regret_bound_validation     # one row per (PDE, probe window)
  field  column              dataType
  -----  ------------------  ----------
         pde                 sc:Text
         k                   cr:Int64    # probe window 10/20/50/100
         n_runs, P           cr:Int64    # runs, candidate policies
         rho_k, rho_p        cr:Float64  # Spearman rho and its p-value
         sigma, delta_gap    cr:Float64  # probe spread, best-2nd gap
         best_policy         sc:Text
         theoretical_bound   cr:Float64
         bound_valid         sc:Boolean
  source: parquet-files-for-config-regret_bound_validation
```

Validation report. `mlcroissant validate` (v1.1.0) on the released `croissant.json` (version 1.3.0) completes with **0 errors and 1 warning**: our JSON-LD `@context` lacks three keys of the reference context (`examples`, `equivalentProperty`, `samplingRate`), none of which the file uses. The raw output is included in the supplementary bundle as `croissant_validation_report.txt`. An earlier draft asserted a clean report before one had been generated; the text above replaces that assertion with the actual output.

N Intended Use, Out-of-Scope Use, and Maintenance Plan

Broader impact. This is a controlled benchmark—no immediate deployment risk. PDE solvers nevertheless feed safety-critical applications (climate, fluid simulation, materials, aerospace); routing heuristics distilled from this synthetic exact-solution benchmark must be OOD-validated before deployment. The release contains only code, configs, protocol specifications, and evaluation JSON artifacts (no trained weights, no human or scraped data); the proportional OOD degradations in Table 4(c) caution against treating hybridization as a robustness guarantee.

Intended use. PINNBENCH is intended for *benchmarking training-policy selection* (i.e. choosing among `pinn_only`, `hybrid_no_gate`, `hybrid_full` and its λ_{physics} variants) in hybrid PINN-operator pipelines, and for controlled PDE-level ablations (causal loss, operator pretraining, gating). The benchmark supports three intended usage modes: *(a)* as a baseline harness for new routing or gating methods (plug in a new `predict(features)` $\rightarrow$ `policy` interface and re-run `paper_a_routing/evaluate_routing_full.py`); *(b)* as a controlled scaffold for PDE-family generalization studies via leave-one-family-out (LOPO-F) cross-validation; and *(c)* as a reference for evaluation hygiene in PDE-solver benchmarking (paired bootstrap CIs, Wilcoxon tests, BH-FDR correction).

Out-of-scope use. PINNBENCH is *not* a substitute for production PDE solving. Users should not transfer our routing heuristics or gate weights directly into *(i)* safety-critical engineering (aerospace, biomedical, structural), *(ii)* noisy-measurement regimes where no exact solution exists, *(iii)* non-exact-solution PDE families (inverse problems with measurement error, PDE-constrained control with stochastic forcing), or *(iv)* high-dimensional PDEs ($d \geq 3$) not represented in the current 13-configuration suite. The benchmark's OOD degradations (Table 4(c)) explicitly caution against treating hybridization as a robustness guarantee.

Maintenance plan. *Versioning:* semver; the camera-ready release is `v1.3.0` (v1.0.0 was the anonymized submission; v1.1.0 and v1.2.0 were earlier camera-ready versions). Future minor versions may add PDE configurations, new routing methods, or new policies; breaking schema changes (result-JSON layout, metric definitions) will bump the major version. *Response SLA:* starting at camera-ready, we commit to a 12-month issue/PR response SLA on the public repository (target: 48 h acknowledgment, 2 weeks for triaged fixes/PR reviews). *Supported environment:*

PyTorch 2.5.1 with Python 3.12 is the baseline; updates may track newer PyTorch releases, and `requirements-lock.txt` pins the environment for reproducing v1.3.0 numbers. *Community contributions:* new PDE configurations are explicitly invited via pull request, provided they implement the `PDEProblem` interface (residual, exact solution, BC/IC) and ship a 10-seed evaluation under the benchmark protocol.

Compute and environmental considerations. All released runs together record about 107 GPU-hours on TITAN Xp GPUs (runs often shared a GPU, which inflates this), at most $\approx$27 kWh at the card's 250 W board power. Repeated probing in routing protocols (5 candidates $\times$ probe budget) increases marginal compute proportionally; users seeking minimum-compute deployment may prefer the physics-loss heuristic, which uses no probing budget beyond what is already required for training.

Contact. Questions and bug reports go to GitHub Issues on the code repository listed in Data, code, and reproducibility, or to the corresponding author at the e-mail address on the first page.

O Evaluation Protocol Checklist

This appendix re-states, in a reviewer-facing format, how each main-text number can be reproduced from a single artifact path or a single deterministic script invocation. The bullets below cover the main-text tables; the other rows of Table 27 (Appendix L) give their artifact and script directly.

- **Table 2 (external positioning)** from the artifacts under `paper_combined/hypino_*`, `paper_combined/pinnacle_subset_5task/`.
- **Table 4 (main $3\times$ slab)** panel (a) hybrid vs. PINN from `rq3_10seed/results.json`, `rq4_10seed/results.json`, `rq1e_kdv_10seed/results.json`, `rq1f_allencahn_10seed/results.json`; panel (b) PINO from `rq_pino_heat1d_10seed/results.json`, `rq_pino_advdiff/results.json`; panel (c) OOD from `rq6_ood*/results.json`.
- **Table 5 (2×2 causal)** from `paper_combined/stage_ablation/*.json` (360 runs); regenerate via `paper_c_causal/stage_ablation.py`.
- **Table 7 (routing)** from `paper_combined/routing_evaluation/results.json` (base methods, 450 runs) and `paper_combined/meta_router/results.json` (PDE-aware router); regenerate via `paper_a_routing/evaluate_routing_full.py` and `paper_combined/evaluate_meta_router.py`.
- **Table 8 (LOPO-F family holdout)** from `paper_combined/routing_family_holdout/results.json` and `paper_combined/meta_router_family_holdout/results.json`; regenerate via `paper_a_routing/evaluate_routing_full.py` and `paper_combined/evaluate_meta_router.py` with `-holdout family`, then `paper_a_routing/routing_tables.py--tablelopof`.
- **Accuracy–regret dissociation** ($77\times$): the fold decisions in `paper_combined/routing_evaluation/results.json` priced with per-(PDE, policy) mean metrics over `paper_a/*/results.json`; `scripts/verify_claims.py` recomputes both regrets (0.0850, 0.0011) and the figure script uses the same code.
- **Diagnostic regret heuristic (44/52)** from `paper_combined/regret_bound_v13/regret_bound_validation.json`; regenerate via `paper_combined/validate_regret_bound.py`.

Traceability guarantee. Each table number is traceable to its artifacts and one deterministic script invocation; see Table 27. No main-text number is aggregated across artifacts without a scripted aggregator committed to the repository.

O.1 Anticipated reviewer questions

Q1. Does the routing memorize PDE identity? Partly. When a PDE's whole family is removed from training, the PDE-aware routers fall from 84.6% to 61.5–69.2% (Table 8), so some of their accuracy comes from same-family configurations. They then route about as well as the training-free `physics_final` heuristic (69.2%), which uses no PDE features at all.

Q2. Why is the HyPINO adapter better on Heat1D? The HyPINO adapter is fine-tuned from a HyPINO foundation checkpoint (500 Adam steps); our hybrid trains from scratch ($200/500/200$ epochs). These are not budget-comparable; we explicitly do not claim to beat HyPINO (Table 4 caption, Table 16).

Q3. Are routing accuracies stable to benchmark composition? For the four base selectors the bootstrap (Appendix E.1) preserves the pre-bootstrap ranking but percentile CIs overlap; the PDE-aware 84.6% router was not bootstrapped, and its LOPO-F drop (Q1) is the relevant stability evidence. Treat point estimates as ranges on this 13-PDE suite.

Q4. Is the gradient-redundancy finding Heat1D-only? No: $\cos \approx 1.0$ holds across 4 PDEs (Heat1D, AdvDiff1D, KdV1D, sharp AllenCahn1D) under both pretraining conditions (Appendix G.1). Causal-weight collapse itself is not observed at the training ϵ (Appendix G.3).

Q5. Is the diagnostic regret heuristic a theorem? No: it is a sub-Gaussian-style sanity check. It holds on 11 of 13 configurations at every window, $k \in \{10, 20, 50, 100\}$ $(44/52)$; of the two Burgers violations, `burgers1d_lownu` is a spread-calibration failure and `burgers1d` is not explained by spread (Appendix I).

Table 27: Claim-to-artifact manifest (main-text tables & numbers). Result paths are rooted at `experiments/results/`; script paths are rooted at `experiments/scripts/`. Note the asymmetric naming convention: *result directories* use short names (`paper_a/`, `paper_c/`, `paper_combined/`), while *script directories* use descriptive suffixes (`paper_a_routing/`, `paper_c_causal/`, `paper_combined/`). This reflects an organic split between result aggregation buckets (short) and the per-paper script collections (descriptive); we preserve it here for fidelity.

Claim (main text)	Artifact (`experiments/results/`)	Regeneration script (`experiments/scripts/`)
Table 4(a) Heat1D, Wave2D hybrid vs. `pinn_only`	`rq3_10seed/results.json`, `rq4_10seed/results.json`	`rq3_model_comparison.py`, `rq4_wave2d_model_comparison.py`
Table 4(a) KdV1D, AllenCahn hybrid vs. PINN	`rq1e_kdv_10seed/results.json`, `rq1f_allencahn_10seed/results.json`	`rq1e_kdv_hybrid.py`, `rq1f_allencahn_hybrid.py`
Table 4(b) PINO (Heat1D, AdvDiff1D)	`rq_pino_heat1d_10seed/results.json`, `rq_pino_advdiff/results.json`	`rq_pino_comparison.py`, `rq_pino_advdiff.py`
Table 4(c) OOD shifts	`rq6_ood*/results.json`, `rq6b_ood_domain/`, `rq6c_ood_ic_shift/`	`rq6_ood_*.py`
Table 5 2×2 stage ablation (360 runs)	`paper_combined/stage_ablation/*.json`	`paper_c_causal/stage_ablation.py`
Gradient cosine ≥ 0.997 (4 PDEs)	`paper_c/gradient_flow_*/gradient_flow_results.json`	`paper_c_causal/gradient_flow_analysis.py`
Temporal error ratio 0.98–672$\times$	`paper_c/temporal_error_analysis/results.json`	`paper_c_causal/temporal_error_analysis.py`
Adaptive-ϵ weight collapse	`paper_combined/adaptive_epsilon/results.json`	`paper_combined/test_adaptive_epsilon.py`
Wang-style single-stage causal (4 PDEs)	`paper_combined/wang_comparison/*_results.json`	`paper_c_causal/wang_comparison.py`
Table 7 base routing (450 runs)	`paper_combined/routing_evaluation/results.json`	`paper_a_routing/evaluate_routing_full.py`
Table 7 PDE-aware 84.6% (11/13)	`paper_combined/meta_router/results.json`	`paper_combined/evaluate_meta_router.py`
Table 8 LOPO-F routing	`paper_combined/{routing,meta_router}_family_holdout/results.json`	`paper_a_routing/evaluate_routing_full.py`, `paper_combined/evaluate_meta_router.py` (`-holdout family`)
Accuracy–regret dissociation (77$\times$)	`paper_combined/routing_evaluation/results.json` + `paper_a/*/results.json`	`scripts/verify_claims.py` (repository root)
Diagnostic regret heuristic (44/52)	`paper_combined/regret_bound_v13/regret_bound_validation.json`	`paper_combined/validate_regret_bound.py`
Epoch-matched causal controls (Table 21)	`paper_c/controls/*/results.json`, `paper_c/control_statistics/results.json`	`paper_c_causal/stage_ablation.py` (`-condition-set controls`), `paper_c_causal/control_statistics.py`
Held-out generalization (3 PDEs, 67%)	`paper_combined/routing_evaluation/holdout.json`	`paper_a_routing/holdout_validation.py`
Scaling $5 \rightarrow 9 \rightarrow 13$ PDEs	`paper_combined/routing_evaluation/scaling_ablation.json`	`paper_a_routing/scaling_and_ablation.py`
HyPINO official native suite	`paper_combined/hypino_official_eval/results.txt`	`paper_combined/evaluate_hypino_official.py`
HyPINO 2-PDE adaptation probe	`paper_combined/hypino_adaptation_2pde/results.json`	`paper_combined/finetune_hypino_*.py`
PINNacle 5-task smoke suite	`paper_combined/pinnacle_subset_5task/results.json`	`paper_combined/run_pinnacle_subset.py`

Table 28: Best policy per PDE under the residual objective (mean over 10 seeds). No single policy dominates across all 13 configurations.

PDE Configuration	Best Policy (Residual)
AdvDiff1D	`hybrid_no_gate`
AdvDiff1D (high Pe)	`pinn_only`
AllenCahn1D	`hybrid_full`
AllenCahn1D (sharp)	`hybrid_full`
Burgers1D	`hybrid_full`
Burgers1D (low ν)	`hybrid_no_gate`
Burgers1D (very low)	`hybrid_no_gate`
Heat1D	`hybrid_full`
KdV1D	`hybrid_no_gate`
NavierStokes2D	`hybrid_full`
ReactDiff1D	`hybrid_full`
ReactDiff1D (stiff)	`hybrid_full`
Wave2D	`pinn_only`

NeurIPS Paper Checklist (Evaluations and Datasets Track)

1. **Claims.** Do the main claims in the abstract and introduction accurately reflect the paper's contributions? **[Yes]** All quantitative claims (7/13 policy split, $77\times$ accuracy–regret dissociation, the epoch-matched causal-loss controls, PDE-aware routing falling from 84.6% to 61.5–69.2% under LOPO-F) are supported by the tables in Sections 5, 6 and 8 and the appendix artifacts in Appendix L.
2. **Limitations.** Did the authors describe the limitations of their work? **[Yes]** Section 9 enumerates five limitations (exact-solution regime, 1D-dominated suite, small policy library, external-method scope, OOD and diagnostic scope); Section 7 reports where the main effect weakens; the routing caveats are in Section 8; out-of-scope use is enumerated in Appendix N.
3. **Theory.** Theoretical results? **[NA]** The paper contributes empirical evaluation and a diagnostic regret heuristic (Appendix I); it does *not* claim a formal theorem.
4. **Experimental reproducibility.** Did the authors fully disclose all information needed to reproduce the main results? **[Yes]** Sections 3 and 4 and Appendix L (Tables 26 and 27) give all hyperparameters, seeds, and the claim-to-artifact manifest; Appendix O gives the reviewer-facing table-regeneration checklist.
5. **Code and data access. [Yes]** Code and benchmark artifacts are public at the URLs in Data, code, and reproducibility (no request required); the Croissant file and its validation output are described in Appendix M; code under MIT, data under Apache-2.0.
6. **Experimental setting/details.** Did the authors specify all training details (data splits, hyperparameters, tuning)? **[Yes]** Section 4 and Appendix L (Table 26); per-script settings are hardcoded in the scripts listed in Table 27, and some result JSONs echo them in a `config` or `hyperparameters` field.
7. **Error bars.** Did the authors report error bars or confidence intervals? **[Yes]** Core tables use 10 seeds (the OOD, adaptive-ϵ and robustness grids use 5; the gradient and temporal diagnostics use one, as stated there) with mean $\pm$ std, paired bootstrap 95% CIs, Wilcoxon p-values, Cohen's d, and BH-FDR correction; accuracy CIs use Wilson intervals (Table 7; Appendix D).
8. **Compute resources. [Yes]** Section 4 and Appendix K: $4\times$NVIDIA TITAN Xp (12 GB); about 107 GPU-hours of recorded training time over all released runs; routing evaluation runs on CPU.
9. **Ethics.** Does the paper conform to the NeurIPS Code of Ethics? **[Yes]** All data is synthetically generated from known PDE solutions; no human subjects, personal data, or copyrighted content; intended and out-of-scope uses enumerated in Appendix N.
10. **Broader impacts.** Does the paper discuss positive and negative societal impacts? **[Yes]** Appendix N, Broader impact paragraph and the out-of-scope-use list.
11. **Safeguards.** Are safeguards described for assets with a high risk of misuse (e.g. models, data)? **[NA]** No pretrained model weights or scraped data are released; only code, configs, and evaluation JSON artifacts.
12. **Licenses.** Are licenses and terms of existing assets properly cited and respected? **[Yes]** FNO, causal PINN, PINO, HyPINO, and PINNacle are cited at point of use (the HyPINO and PINNacle code we ran is MIT-licensed); PyTorch, scikit-learn, NumPy and SciPy are named with versions in Appendix K; our code is released under MIT and the benchmark data under Apache-2.0.
13. **Asset release.** Is the new asset (benchmark) well-documented and distributed responsibly? **[Yes]** The submission includes PDE specifications (Appendix A), baseline-protocol specifications (Appendix B), metric definitions (Appendix C), a claim-to-artifact manifest (Appendix L), a reproducibility manifest (Appendix L), Croissant metadata (Appendix M), and an intended-use / maintenance-plan statement (Appendix N).
14. **Crowdsourcing. [NA]** No crowdsourcing is used.
15. **IRB. [NA]** No human-subjects research.
16. **LLM usage. [Yes]** A generative AI coding assistant was used to implement library and experiment code, design and critique experiments, propose hypotheses, formulate the assumptions of the regret diagnostic, interpret results, and draft and edit text. The authors reviewed all such work. Every headline number is recomputed from the released JSONs by `verify_claims.py`, and every reference was checked against a primary bibliographic record (DOI registry, OpenReview, or the publisher's page).