LEARN OR REFINE? PREDICTING WHEN NEURAL PDE CORRECTION IS WORTH TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned error correctors repair the output of a cheap classical PDE solver, but training one costs high-fidelity data and optimization. Before paying that cost, one would like to know whether the corrector will beat the classical alternative of refining the grid. Accuracy against a fixed reference cannot answer this, because refinement spends the same resource the corrector spends on training data. We formulate the question as a prospective matched-cost decision: a corrector is worth training only if its achievable error beats both a ridge control and the best refinement affordable within its own budget. The hard part is that the achievable learned error is unknown before training. We show that the pilot-to-full remaining learning gain is strongly family-dependent, so a fixed pilot-to-full calibration does not transfer reliably across families; but the gain is predictable from the pilot's early learning progress. This yields a two-feature estimator and a Learn-or-Refine rule costing about 15% of full training. In leave-one-family-out evaluation over 74 prior regime$\times$architecture cases it has the lowest cross-family decision regret among competing estimators; after freezing the estimator it decides correctly on 9 of 10 regimes of an unseen Allen–Cahn family with zero decision regret. The method decides whether a specified corrector recipe is worth completing relative to matched-cost classical refinement.

1 INTRODUCTION

A learned PDE error corrector maps the output of a cheap, low-accuracy classical solver toward a high-fidelity reference solution. Correctors are now common (Bar-Sinai et al., 2019; Um et al., 2020; Wei et al., 2025; Li et al., 2020; Huang & Perdikaris, 2026), and inference is cheap once the model is trained. Training is not free: it requires many high-fidelity solves for training data, plus optimization. A practitioner facing a new problem must decide whether to make that investment before making it.

Consider a concrete scenario. A researcher has a 2D advection–diffusion problem and a budget of 1000 deployment rollouts. Training a corrector requires 256 high-fidelity reference solves for training data, plus the optimization itself. The same 256 reference solves could instead pay for running the classical solver on a finer grid for every rollout, or for Richardson extrapolation between two resolutions. Which use of the budget gives lower error? The answer depends on the problem: for some parameters the corrector wins by a large margin; for others the classical refinement is better; and for many the two are within noise. The researcher cannot know which case they are in without training the corrector — which is exactly the cost they are trying to avoid.

The natural evaluation — comparing the trained corrector's error to the coarse solver's error — cannot answer this question. The same compute that pays for training data could instead be spent refining the grid or extrapolating between resolutions, and classical refinement is often effective at matched cost (Zhang et al., 2026; McGreivy & Hakim, 2024). The decision is not "is the corrector accurate?" but "is the corrector better than what the same budget buys classically?"

The answer depends on the corrector's achievable error at full training, which is unknown before training. The classical side is cheap to measure; the learned side is exactly what one is trying to avoid paying for. A fixed pilot-to-full calibration — train a tiny pilot, measure its error, map it to the full error — works within one problem family but fails under cross-family transfer. We show the reason: the ratio of pilot error to full error, the remaining learning gain, varies strongly across PDE

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

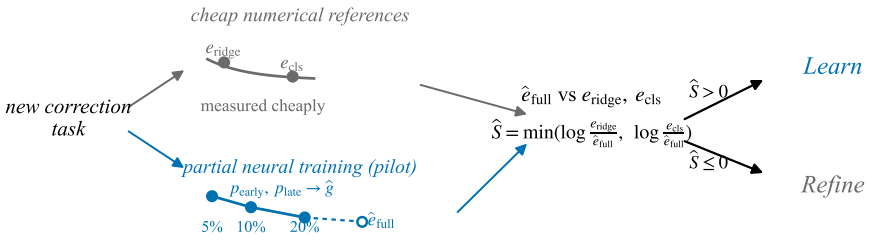

Figure 1: **Learn-or-Refine.** Cheap numerical references (e_{ridge}, e_{cls}) define the payoff frontier. A partial neural training trajectory ($5\%, 10\%, 20\%$ of the full epoch budget) yields early progress ($p_{\text{early}}, p_{\text{late}}$), which estimates the remaining learning gain $\widehat{g}$ and hence the full-training error $\widehat{e}_{\text{full}}$. Comparing $\widehat{e}_{\text{full}}$ with the numerical references gives the payoff $\widehat{S}$ and the decision to complete training or refine classically.

families. The more robust observation is that the gain is visible in the pilot's early learning progress: a pilot that is still improving quickly has more room to improve than one that has plateaued.

Our method, Learn-or-Refine, trains a pilot corrector on 10% of the training pairs for 20% of the full epoch budget, reads its early progress, maps it to an estimated remaining gain, and decides Learn or Refine by comparing the predicted full-corrector error against the cheap classical and ridge references (Figure 1). It uses no PDE-family or dimension labels. Because real problem sets are class-imbalanced, we evaluate primarily by decision regret, the payoff lost relative to an oracle.

Our main findings are:

- We formulate whether to train a learned PDE corrector as a prospective matched-cost algorithm-selection problem, with a payoff margin $S^* = \min(\log \frac{e_{\text{ridge}}}{e_{\text{full}}}, \log \frac{e_{\text{cls}}}{e_{\text{full}}})$ (§3).
- We show that the pilot-to-full remaining gain is family-dependent, so absolute pilot calibration does not transfer; but early learning progress is a transferable signal (§3.3, §6.4).
- We validate the resulting Learn-or-Refine rule by leave-one-family-out over 74 regime×architecture cases and a frozen prospective confirmation on an unseen Allen–Cahn family (§6.5).

2 Related Work

Learned correctors and hybrid solvers. Learned error correction for PDEs spans learned discretizations (Bar-Sinai et al., 2019), solver-in-the-loop training (Um et al., 2020), indirect neural correctors (Wei et al., 2025), neural operators (Li et al., 2020), and training-free stabilization (Huang & Perdikaris, 2026). These works propose corrector architectures: they show that a neural network can learn to correct a coarse solver, and they evaluate the corrector on accuracy and speed. Our decision rule selects when to train them: it decides whether the corrector will beat the classical alternative of refining the grid, before the corrector is trained.

Cost-aware evaluation. McGreivy & Hakim (2024) critique weak baselines and suggest an amortization quantity; Zhang et al. (2026) formalize breakeven complexity, the number of solves before a learned solver is cost-effective relative to an error-matched classical solver; de Hoop et al. (2022) study the cost–accuracy trade-off in operator learning. These evaluate the trade-off after training: they measure the cost-effectiveness of an already-trained solver. Our contribution is the prospective decision: predicting, before training, whether the investment pays off. Breakeven complexity asks "was it worth it?"; we ask "will it be worth it?"

Learning-curve extrapolation and multi-fidelity selection. Partial training curves are used to predict final performance and to allocate resources among learned configurations (Domhan et al., 2015; Mohr & van Rijn, 2024; Li et al., 2017). Those methods decide *which learned configuration to keep* or *when to stop it*. Learn-or-Refine makes a different decision from the same kind of signal:

whether to complete learning at all, or switch the budget to a numerical refinement arm with a different cost–accuracy scaling law.

Algorithm selection. Per-instance algorithm selection (Xu et al., 2008) chooses among solvers from cheap features. Our problem is a mechanism-grounded instance: Learn-or-Refine at matched cost, with a signal from the corrector's own dynamics.

Neural/classical solver routing. A related line of work routes between neural and classical solvers at inference time, assuming the neural solver is already trained. We instead route before training: whether to invest in it at all.

3 The Learn-or-Refine problem

3.1 Matched-cost formulation

We consider residual correction with a coarse solver S_h on grid h and a reference solver $S_{h_{\mathrm{ref}}}$ on $h_{\mathrm{ref}} = h/8$. A corrector C_θ is trained on pairs $(u_h, u_{\mathrm{ref}} - u_h)$. Its end-to-end cost has three parts: generating n_{train} training pairs, training, and per-rollout inference. Classical refinement spends the same resource per rollout: solving on $h/2$ or $h/4$, or Richardson-extrapolating between them (Richardson & Gaunt, 1927; Trottenberg et al., 2001). For N deployment rollouts the learned total cost is

$$C_{\mathrm{learn}} \;=\; n_{\mathrm{train}}\, c_{\mathrm{ref}} \;+\; c_{\mathrm{train}} \;+\; N\,(c_h + c_{\mathrm{infer}}), \tag{1}$$

where c_{ref} is one reference solve and c_h one coarse solve. A classical arm is affordable if its per-rollout cost times N does not exceed C_{learn}. Unless stated otherwise we use $N = 1000$; §6.6 reports sensitivity.

The training-data cost is the hidden confounder. In our experiments, generating 256 reference solves costs 286–501 seconds, while a single fine solve costs 0.02–0.29 seconds: the training-data cost is three to four orders of magnitude larger than one fine solve. The standard protocol compares expensive training against cheap inference and omits the refinement the same budget would buy; the matched-cost formulation restores it.

3.2 The payoff margin

Let e_{full} be the trained corrector's error, e_{ridge} the error of a ridge corrector, and e_{cls} the error of the best affordable classical arm. Define

$$a = \log \frac{e_{\mathrm{ridge}}}{e_{\mathrm{full}}}, \qquad b = \log \frac{e_{\mathrm{cls}}}{e_{\mathrm{full}}}, \qquad S^* = \min(a, b). \tag{2}$$

The corrector is worth training iff $S^* > 0$: a asks whether it beats a trivial learned baseline, and b whether it beats the best classical use of its own budget. In our 12-regime audit $S^* > 0$ separates all outcomes; neither margin alone suffices.

The ridge margin a is not arbitrary. The ridge control fits a per-grid-point linear map from local stencil features $[1, u, u_x, u_{xx}]$ to the residual. If this cheap linear map already corrects near-exactly, the required correction is a low-order local map, which is precisely the structure Richardson extrapolation cancels analytically; a neural corrector would then be regressing, from data and with estimation error, a function classical analysis knows in closed form. If the ridge fails to correct, the correction is non-linear in the local features and learning may be necessary.

3.3 Remaining learning gain

The classical side of equation 2 is cheap to measure. The unknown is e_{full}. A cheap pilot corrector trained at a fraction of the budget gives a pilot error e_{pilot}, and the question is how to estimate e_{full} from it. Define the remaining learning gain

$$g = \log \frac{e_{\mathrm{pilot}}}{e_{\mathrm{full}}}, \tag{3}$$

so that $e_{\text{full}} = e_{\text{pilot}} e^{-g}$. A fixed pilot-to-full calibration maps e_{pilot} to e_{full} through a single global relation — constant-gain or log-linear — fit once on a development family. We show in §6.4 that g is strongly family-dependent (mean from 0.06 to 1.12 across families), so any such global relation is benchmark-specific.

The transferable observation is that g is encoded in the pilot's early learning progress. Let $e_{5\%}, e_{10\%}, e_{20\%}$ be the pilot's error at $5\%, 10\%, 20\%$ of the full epoch budget, and define

$$p_{\text{early}} = \log \frac{e_{5\%}}{e_{10\%}}, \qquad p_{\text{late}} = \log \frac{e_{10\%}}{e_{20\%}}. \tag{4}$$

A pilot that is still improving quickly has more remaining gain than one that has plateaued. The estimator is a two-feature linear model,

$$\widehat{g} \;=\; 0.2394 \;+\; 0.4893\, p_{\text{early}} \;+\; 1.1240\, p_{\text{late}}, \qquad \widehat{e}_{\text{full}} = e_{\text{pilot}}\, e^{-\widehat{g}}, \tag{5}$$

fit once on historical regimes and frozen before prospective evaluation. We also evaluated pilot-capacity scaling and a residual ridge-gap signal; neither improved cross-family regret (§6.4).

3.4 The decision rule

Learn-or-Refine computes $(e_{\text{ridge}}, e_{\text{cls}})$ and $\widehat{e}_{\text{full}}$, forms

$$\widehat{S} = \min\left(\log \frac{e_{\text{ridge}}}{\widehat{e}_{\text{full}}},\; \log \frac{e_{\text{cls}}}{\widehat{e}_{\text{full}}} \right), \tag{6}$$

and chooses **Learn** iff $\widehat{S} > 0$, else **Refine**. The pilot is an 8-channel CNN on 10% of training pairs for 20% of the full epoch budget; the decision costs about 15% of full training.

3.5 Decision regret

Because real problem sets are class-imbalanced, accuracy can be gamed by always choosing the majority class, so we use decision regret as the primary metric. The decision is between two arms — complete the corrector or refine — and the ground-truth winner is the gate winner of equation 2: Learn only if the corrector beats both the ridge control and refinement. A matching decision incurs no regret; a wrong one incurs $(e - e^*)/e^*$, where $e^* = \min(e_{\text{full}}, e_{\text{cls}})$ is the better deployable arm. The ridge is thus an eligibility gate, not a third arm, and drops out of the operational comparison. (The one regime where the gate winner and the better arm differ, AC11, is zero-margin.)

3.6 A worked example

Consider a 3D advection–diffusion problem with $\nu = 0.02$, $N_c = 16$, and $N = 1000$ deployment rollouts. The classical frontier gives $e_{\text{ridge}} = 0.163$ and $e_{\text{cls}} = 0.0558$ (Richardson on $h/2, h/4$). A pilot corrector trained on 26 pairs for 20% of the full epoch budget reaches $e_{\text{pilot}} = 0.057$ with early progress $p_{\text{early}} = 0.637$ and $p_{\text{late}} = 0.881$. The estimator gives $\widehat{g} = 0.239 + 0.489(0.637) + 1.124(0.881) = 1.54$, so $\widehat{e}_{\text{full}} = 0.057\, e^{-1.54} = 0.0122$. The predicted margin is $\widehat{S} = \min(\log(0.163/0.0122), \log(0.0558/0.0122)) = \min(2.59, 1.52) = 1.52 > 0$, so the decision is Learn. The full corrector reaches $e_{\text{full}} = 0.0073$, so the true margin is $S^* = \min(\log(0.163/0.0073), \log(0.0558/0.0073)) = 2.03 > 0$: the decision is correct, and the corrector beats the classical frontier by a factor of 7.6 in error.

4 Three correction regimes

The two margins reveal three distinct correction regimes across the audited problems (Figure 2). (i) *Local-linear*: the ridge control matches the CNN (1D Burgers smooth, 2D Darcy). The correction is a low-order local map, Richardson cancels it analytically, and classical wins. (ii) *Non-linear-but-learnable*: the CNN beats the ridge (the advection–diffusion and Navier–Stokes problems). Learning captures real non-linear structure, but the corrector wins only if refinement is not near-exact at budget, which holds in the advection–diffusion regimes where the solver converges slowly. (iii) *Global-phase*: both the ridge and the CNN fail (Helmholtz). The pollution error is a global phase

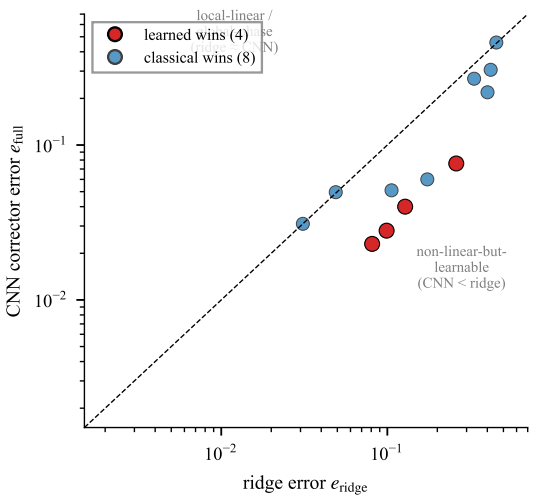

Figure 2: Learned-corrector error vs. few-parameter-ridge error across regimes (log-log). Points on the dashed diagonal (ridge = CNN) are regimes where the correction is local-linear or global-phase and classical wins. Points off the diagonal (CNN $<$ ridge) are regimes where learning captures non-linear structure. The four learned wins (red) are the only regimes where the CNN beats the ridge *and* refinement is not near-exact at matched cost.

drift that a local model cannot correct; neither learned nor local classical methods help, and only refinement works. Together, the three regimes account for the observed outcome table and give the learned-versus-classical boundary a mechanistic interpretation.

The residual analysis of the canonical win confirms this. In 2D advection–diffusion the true residual is large (relative norm 0.66), low-frequency (99% of its energy below $k = 4$), and non-linear in the local features: the CNN predicts it almost perfectly (correlation 0.998) while the ridge cannot (relative L_2 0.19, against the CNN's 0.05). With a slowly converging first-order upwind solver, even the $h/4$ refinement leaves error 0.096. All three conditions hold in this regime and in the other slow-refinement advection–diffusion wins.

5 Why early progress predicts remaining gain

The remaining learning gain g measures how much the full corrector improves over the pilot. A large g means the pilot is far from the full corrector's achievable error; a small g means the pilot is already close. The gain depends on the PDE family because different families present different learning challenges: advection–diffusion correctors learn to remove numerical diffusion, Fisher–KPP correctors learn to sharpen reaction fronts, and Helmholtz correctors struggle with dispersion errors that are hard to correct locally.

The early learning progress signals p_{early} and p_{late} capture the slope of the pilot's learning curve at two points. A steep early slope (p_{early} large) indicates that the corrector is still learning rapidly at 10% of the budget; a steep late slope (p_{late} large) indicates that learning continues at 20%. Both predict a large remaining gain. A pilot that has plateaued by 20% has little room to improve, and its remaining gain is small.

Figure 3 illustrates this on three representative regimes. The advection–diffusion pilot is still improving rapidly at 20% of the budget, and the full corrector reaches a much lower error ($g = 2.05$). The Burgers pilot has slowed but is still improving ($g = 0.20$). The Helmholtz pilot has nearly plateaued, and the full corrector barely improves ($g = -0.13$). The early slope is a reliable indicator of how much room remains.

This is not the same as learning-curve extrapolation (Domhan et al., 2015), which predicts the final validation error of a single training run. We predict the remaining gain of a corrector relative to classical refinement, which is a different quantity. The gain is not the final error; it is the ratio of

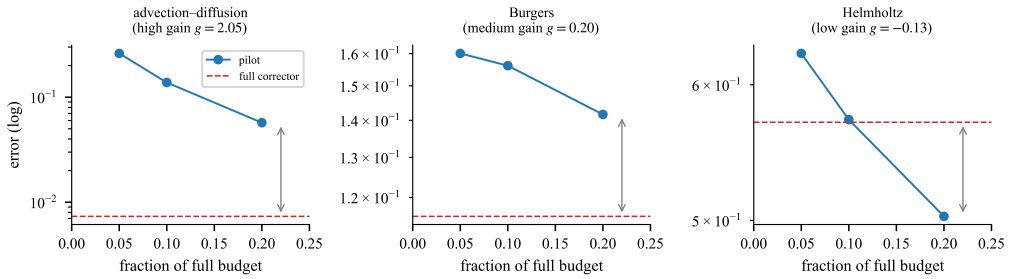

Figure 3: Pilot learning curves (5%, 10%, 20% of full epoch budget) and full-corrector error for three representative regimes. The remaining gain $g = \log(e_{\text{pilot}}/e_{\text{full}})$ is the gap between the pilot error at 20% and the full-corrector error. A steep early slope predicts a large remaining gain.

pilot error to full error, and it is this ratio that determines whether the corrector beats the classical frontier.

6 Experiments

6.1 PDE families and correctors

The decision-rule development and validation span five PDE families with distinct numerical behavior. Advection–diffusion (1D/2D/3D) uses a first-order upwind scheme; the corrector learns to remove numerical diffusion. Viscous Burgers (1D and 2D) uses the same upwind scheme; the corrector learns to sharpen shocks. High-wavenumber Helmholtz (1D) is dispersion-dominated; the corrector struggles because dispersion errors are non-local. Fisher–KPP reaction–diffusion (1D/2D) uses a second-order central Laplacian; the corrector learns to sharpen reaction fronts. Allen–Cahn phase-field (2D) uses a second-order central Laplacian with a stiff reaction term; the corrector learns to resolve interphase fronts.

Each family uses a coarse→reference hierarchy ($h_{\text{ref}} = h/8$) with classical arms $\{h/2, h/4, \text{Richardson}(h, h/2), \text{Richardson}(h/2, h/4)\}$ and a ridge control on local stencil features. The corrector is a small CNN (24-channel 2D / 16-channel 3D / 32-channel 1D) with 3×3 (2D/3D) or 5×5 (1D) circular convolutions, GELU activations, and Adam (lr 10^{-3}, or 2×10^{-3} for 3D). The pilot corrector is an 8-channel CNN with the same architecture at reduced width. A U-Net variant (channel-matched additive outer skip) is used for the architecture-transfer check. Full training uses all 256 training pairs, the full epoch budget, no early stopping, and 3 seeds; the reported full-corrector error is the mean across seeds. The pilot uses 26 pairs (10%), 20% of the epoch budget, and seed 1000. The classical arms use the same solver at finer resolution; Richardson extrapolation combines two resolutions to cancel the leading error term. The ridge control fits a linear model on local stencil features; it is a trivial learned baseline that any useful corrector should beat. Full problem definitions and architecture details are in Appendices A and B.

6.2 Development, selection, and prospective evaluation

We separate three kinds of evidence. *Development* evidence (the 12-regime audit and a first set of pilot studies) motivated the mechanism and the method. *Model-selection* evidence is a leave-one-PDE-family-out (LOFO) analysis over 74 regime×architecture cases across four families, used only to choose the estimator. *Prospective* evidence is the frozen Allen–Cahn evaluation, performed only after the estimator was frozen, on a family unseen during fitting and selection. The Allen–Cahn evaluation is the headline result.

6.3 Baselines and metrics

We compare against six baselines spanning the design space. *Always Learn* and *Always Classical* are trivial policies that always choose one arm; they establish the floor for any decision rule. The *classical-only probe rule* is a local-linearity screen plus a refinement-affordability check: it predicts

Table 1: Leave-one-family-out decision regret over 74 regime×architecture cases (4 families). Micro = pooled over cases; macro = mean of per-family means. Lower is better.

Method	micro regret	macro regret	accuracy
Always Learn	25.28	28.68	0.378
Always Classical	0.374	0.217	0.622
raw pilot ($\widehat{e}_{\mathrm{full}} = e_{\mathrm{pilot}}$)	0.370	0.214	0.635
M0c constant-gain (refit)	0.084	0.054	0.851
B2-d10 absolute calibration (frozen)	0.070	0.044	0.878
Learn-or-Refine (M1b)	**0.063**	**0.038**	**0.892**

Learn only if the coarse error is large and refinement is not affordable within the budget. This is the natural pre-training heuristic. The *raw pilot* uses $\widehat{e}_{\mathrm{full}} = e_{\mathrm{pilot}}$ directly, ignoring the remaining gain entirely. *M0c* is a constant-gain calibration ($\widehat{g}$ is the pooled mean gain, refit per fold). *Estimator A* is a generic learning-curve extrapolation that uses the full dataset to predict the final error; it costs $\sim 79\%$ of full training. *B2-d10* is the frozen absolute pilot-to-full calibration from our earlier work: a log-linear (power-law) map from e_{pilot} to e_{full}, fit once on advection–diffusion and never refit. The progression is classical-only → raw pilot → absolute calibration → learning dynamics. Table 1 evaluates the pilot-based estimators by LOFO; the probe and Estimator A appear in Table 2. The primary metric is decision regret; we also report accuracy, confusion counts, and decision cost.

6.4 Historical cross-family validation

On the original 12-regime audit the learned corrector wins in a coherent advection–diffusion family (4 of 12) and $S^* > 0$ separates all 12. The two cheap classical-only probes (the local-linearity screen and the refinement-affordability check) are correct on only 1 of 3 held-out regimes, both errors false negatives.

The remaining gain g is strongly family-dependent (Figure 4): mean g is $+0.82$ for advection–diffusion, $+1.12$ for Fisher–KPP, $+0.29$ for Burgers, and $+0.06$ for Helmholtz. A frozen calibration fit on advection–diffusion therefore mispredicts the gain elsewhere: on a held-out transfer set of 33 Fisher–KPP and 2D Burgers cases it is beaten by the uncalibrated raw pilot (0.879 vs 0.970 accuracy), and its errors are confident false positives on the corrector-weak family. (That transfer set is classical-dominated, so the raw pilot's 0.970 there matches the majority policy; the calibration's failure is that it does *worse* than this trivial reference, not that the raw pilot is strong.) The oracle decomposition confirms the payoff boundary itself is sound: with the true e_{full}, the cheap classical information alone decides $33/33$ correctly.

Table 1 reports the LOFO cross-family regret; the macro (per-family-mean) aggregate is the primary generalization metric, with the pooled micro value as secondary detail. The progress-based estimator (M1b) has the lowest regret among all candidates. Capacity scaling and the residual gap did not help; the residual gap hurt because its correlation with g flips sign across families. Figure 5 shows the per-family breakdown: M1b's advantage is concentrated on the Burgers family, where the frozen absolute calibration produces confident false positives.

The payoff geometry is not an artifact of the small CNN. A stronger FNO corrector enlarges the learned margin in the canonical learned-win regime ($1.84\times$ vs the CNN's $1.19\times$) and still loses by $6\times$ in the canonical classical-win regime; capacity and data-budget sweeps leave the classical losses unchanged (Appendices J–K).

6.5 Frozen prospective evaluation on Allen–Cahn

After freezing the estimator equation 5, we evaluate it on 2D Allen–Cahn $u_t = \varepsilon \Delta u + (u - u^3)/\varepsilon$, a phase-field family unseen during fitting and selection, with sharp interphase fronts. The regime set (10 regimes, $\varepsilon \in [0.002, 0.1]$, $N_c \in \{16, 32\}$) was frozen before evaluation; five regimes were removed or replaced for compute infeasibility before any of their outcomes were observed (Appendix P). Ground truth: 2 Learn (sharp fronts), 8 Refine.

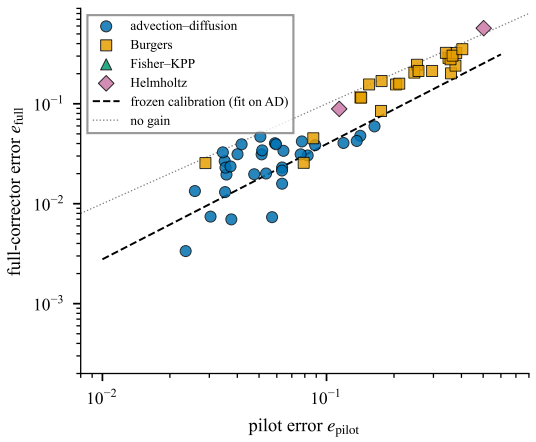

Figure 4: Why absolute pilot calibration is insufficient. Pilot error vs full-corrector error for the 74 historical cases, colored by PDE family. The frozen calibration (dashed, fit on advection–diffusion) fits that family but misses the others: at a similar pilot error, Burgers retains a higher full error (small remaining gain) while Fisher–KPP reaches a lower one (large gain). The dotted line is no remaining gain ($e_{\text{full}} = e_{\text{pilot}}$).

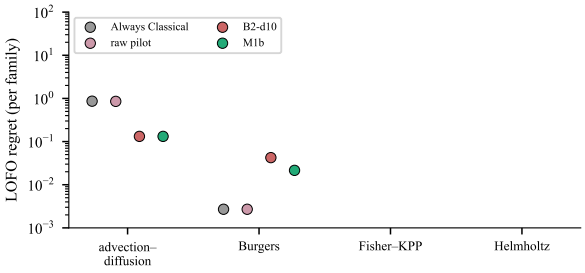

Figure 5: LOFO regret by family for the four main estimators. M1b's advantage is concentrated on Burgers, where the frozen absolute calibration (B2-d10) produces confident false positives.

Learn-or-Refine decides **9 of 10** correctly (accuracy 0.900, Wilson 95% $[0.60, 0.98]$), with **zero decision regret** and a decision cost of 14.7% of full training (Table 2, Figure 6). It catches both sharp-front Learn cases (AC04, AC14), where the corrector resolves the interphase front better than any affordable classical refinement, and its only error is a single false positive on AC11 — a regime where the payoff margin is nearly zero ($e_{\text{full}}, e_{\text{cls}}, e_{\text{ridge}}$ all within 5%), so the decision is arbitrary and the regret is zero. Estimator A also reaches 0.900 with zero regret, but at $\sim 79\%$ of full training cost, versus 14.7% for our method: Estimator A uses the full dataset to extrapolate the learning curve, while our method uses only 10% of the data and reads the early slope.

The frozen absolute calibration (B2-d10) reaches 0.800 on Allen–Cahn, with regret 0.052. It also catches both Learn cases, but it over-estimates the remaining gain on this family — its calibration was fit on advection–diffusion, where the gain is larger — and so predicts Learn too readily, making two false-positive calls. One of them (AC15) is costly (regret 0.523): B2-d10 predicts the corrector will improve enough to beat refinement, but it does not. The classical-only probe rule shows the same profile (0.800, regret 0.052). The raw pilot reaches 0.900 but makes the opposite error, missing one Learn case (AC04) because it ignores the remaining gain entirely. Learn-or-Refine is the only method that both catches every Learn case and avoids the costly false positive.

The 10 regimes span $\varepsilon \in \{0.002, \ldots, 0.1\}$ at $N_c \in \{16, 32\}$, covering sharp-front ($\varepsilon \leq 0.01$) and smooth ($\varepsilon \geq 0.02$) cases. The sharp-front regimes are contested: the corrector can learn to resolve the interphase front that refinement cannot reach cheaply, while the smooth regimes are classical-leaning. M1b decides every smooth regime correctly and misclassifies only one sharp-front regime, the zero-margin AC11.

Table 2: Frozen prospective results on the unseen Allen–Cahn family (10 regimes). Regret is the primary metric. Decision cost is the mean fraction of full training cost.

Method	accuracy	mean regret	max regret	decision cost
Always Learn	0.200	35.1	125.2	0%
Always Classical	0.800	0.040	0.203	0%
classical-only probe	0.800	0.052	0.523	4.9%
raw pilot	0.900	0.020	0.203	14.7%
B2-d10 absolute calibration	0.800	0.052	0.523	14.7%
Estimator A (learning curve)	0.900	0.000	0.000	79%
Learn-or-Refine (M1b)	**0.900**	**0.000**	**0.000**	**14.7%**

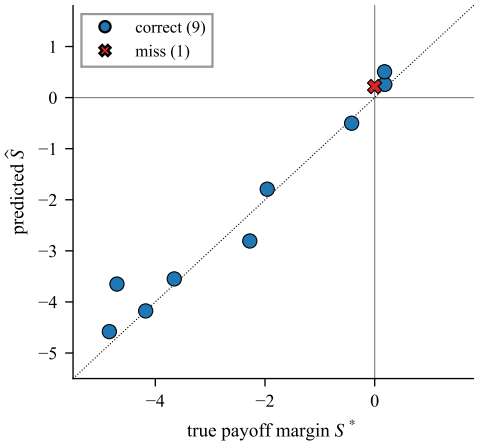

Figure 6: True payoff margin S^* vs predicted $\widehat{S}$ on the unseen Allen–Cahn family. Points in the upper-right and lower-left quadrants are correct; the single near-boundary miss (red) is a zero-margin regime.

6.6 Sensitivity and failure analysis

Deployment volume. The matched-cost decision depends on the deployment volume N through equation 1: recomputing for $N \in \{10, 100, 10^3, 10^4, 10^5\}$, several regimes flip from Refine to Learn as N grows and the up-front cost amortizes, and none flip back (Appendix N).

A failure case: the very strong corrector. The method's one catastrophic historical failure is a well-resolved 2D advection–diffusion regime (D8) where the full corrector reaches a very low error ($e_{\text{full}} = 0.0034$): the true remaining gain is $g = +1.95$ but the pilot has nearly plateaued ($\widehat{g} = +0.36$), so the method predicts Refine while the truth is Learn (regret 2.88). The frozen absolute calibration fails identically. When the achievable error is far below the pilot's reach, the pilot carries no signal of it, so the gain is underestimated precisely when the corrector is unusually strong (Appendix O).

Protocol amendments. Three fine-grid and two large-ε Allen–Cahn regimes were removed or replaced before any of their outcomes were observed, because the stiff diffusion timestep made their reference solves infeasible (Appendix P).

7 Discussion and limitations

Learn-or-Refine evaluates whether a specified corrector recipe is worth completing relative to matched-cost refinement, for residual-correction PDE problems with a coarse→reference hierarchy. We test five PDE families; the estimator's coefficients are a pooled fit, transfer beyond these families is untested, and near-boundary labels are protocol-sensitive. D8 shows the main failure mode: unusually strong late learning is invisible to the pilot.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for providing feedback on research methodology and experiments, providing feedback on mathematical arguments, and interpreting results. We have not used generative AI tools for generating synthetic data sets, formulating mathematical claims or implementing methods, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for literature discovery, brainstorming, drafting and editing parts of the manuscript, editing software code and figures, and simulated peer-review feedback. We have reviewed all AI-assisted work and AI-assisted code was re-run with its outputs checked against the experimental artifacts reported in this paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

All design choices were frozen in protocol files before the corresponding evaluations; the final estimator coefficients in equation 5 were frozen before the Allen–Cahn family was generated or evaluated. Ground-truth labels use a deterministic full-budget protocol (full-width corrector, all training pairs, full epoch budget, no early stopping, three seeds). Seeds are fixed for Python, NumPy, and PyTorch, and the pilot uses seed 1000.

The supplementary material contains the protocol configurations, pilot and full-budget evaluation records, frozen estimator information, solver and analysis code, and the experimental records used for the reported results. The supplied scripts reproduce the principal matched-cost comparisons, leave-one-family-out evaluation, sensitivity analyses, and prospective Allen–Cahn decision results.

REFERENCES

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019. doi: 10.1073/pnas.1814058116.

Maarten V. de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. The cost-accuracy trade-off in operator learning with neural networks. *arXiv preprint arXiv:2203.13181*, 2022.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3460–3468, 2015.

Xinquan Huang and Paris Perdikaris. Physicscorrect: A training-free approach for stable neural pde simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. arXiv:2507.02227.

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, 2017.

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

Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6 (10):1256–1269, 2024. doi: 10.1038/s42256-024-00897-5. arXiv:2407.07218.

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

Lewis Fry Richardson and J. Arthur Gaunt. The deferred approach to the limit. part i. single lattice. part ii. interpenetrating lattices. *Philosophical Transactions of the Royal Society A*, 226:299–361, 1927. doi: 10.1098/rsta.1927.0008.

Ulrich Trottenberg, Cornelis W. Oosterlee, and Anton Schüller. *Multigrid*. Academic Press, San Diego, 2001.

Kiwon Um, Robert Brand, Yun (Raymond) Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6111–6122, 2020. arXiv:2007.00016.

Hao Wei, Aleksandra Franz, Björn Malte List, and Nils Thuerey. Inc: An indirect neural corrector for auto-regressive hybrid pde solvers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2511.12764.

Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Satzilla: Portfolio-based algorithm selection for sat. *Journal of Artificial Intelligence Research*, 32:565–606, 2008.

Yijing Zhang, Nicholas Roberts, Tanya Marwah, and Mikhail Khodak. Breakeven complexity: A new perspective on neural partial differential equation solvers. *arXiv preprint arXiv:2605.15399*, 2026.

A Full problem definitions

Table 3: PDE families, discretizations, and verification. All use a coarse→reference hierarchy with $h_{\mathrm{ref}} = h/8$ and periodic boundaries. Convergence orders p are measured against a further-refined reference before any learning.

family	equation	discretization	verified
advection–diffusion	$u_t + c \cdot \nabla u = \nu \Delta u$	upwind (1st) + central Δ	$p \approx 1$, ref. conv.
Burgers	$u_t + u u_x = \nu u_{xx}$ (1D, 2D scalar)	upwind + central Δ	$p \approx 1$, ref. conv.
Helmholtz	$\Delta u + k^2 u = f$	2nd-order central	dispersion-dominated
Fisher–KPP	$u_t = D\Delta u + ru(1-u)$	2nd-order central Δ	$p \approx 2$, bounded $[0,1]$
Allen–Cahn	$u_t = \varepsilon \Delta u + (u - u^3)/\varepsilon$	2nd-order central Δ	$p \approx 2.2$–2.6, $u \in [-1,1]$

For each family, the governing equation, discretization, boundary/initial conditions, and parameter ranges are given in Table 3. All problems are periodic on the unit domain with a coarse→reference hierarchy ($h_{\mathrm{ref}} = h/8$). Each solver was verified independently before any learning: measured convergence order against a further-refined reference, reference-convergence gap, and physical admissibility (boundedness for Fisher–KPP and Allen–Cahn).

A.1 Advection–diffusion

The equation is $u_t + c \cdot \nabla u = \nu \Delta u$ with periodic boundary conditions on $[0,1]^d$ ($d = 1, 2, 3$). The coarse solver uses first-order upwind for advection and second-order central differences for diffusion. The reference solver uses the same scheme on a grid $8\times$ finer. Parameters: $\nu \in [0.01, 0.05]$, $c = (1.0, 0.6, 0.4)$ (truncated for 2D/1D), $N_c \in \{16, 32, 48\}$, $T \in [0.3, 0.4]$. The corrector learns to remove the numerical diffusion introduced by the upwind scheme.

A.2 Viscous Burgers

The equation is $u_t + u u_x = \nu u_{xx}$ (1D) or $u_t + u \cdot \nabla u = \nu \Delta u$ (2D scalar) with periodic boundary conditions. The coarse solver uses first-order upwind for advection and second-order central differences for diffusion. Parameters: $\nu \in [0.01, 0.05]$, $N_c \in \{16, 32\}$, $T \in [0.3, 0.4]$. The corrector learns to sharpen the numerical diffusion at shocks.

A.3 High-wavenumber Helmholtz

The equation is $\Delta u + k^2 u = f$ on $[0, 1]$ with periodic boundary conditions. The coarse solver uses second-order central differences. The reference solver uses the same scheme on a grid $8\times$ finer. Parameters: $k \in [10, 20]$, $N_c \in \{32, 64\}$. The problem is dispersion-dominated; the corrector struggles because dispersion errors are non-local.

A.4 Fisher–KPP reaction–diffusion

The equation is $u_t = D\Delta u + ru(1 - u)$ with periodic boundary conditions on $[0, 1]^d$ ($d = 1, 2$). The coarse solver uses second-order central differences for the Laplacian and explicit Euler for the reaction term. The reference solver uses the same scheme on a grid $8\times$ finer. Parameters: $D \in [0.01, 0.1]$, $r \in [1, 10]$, $N_c \in \{16, 32\}$, $T \in [0.3, 0.4]$. The corrector learns to sharpen the reaction front. Solutions are bounded in $[0, 1]$.

A.5 Allen–Cahn phase-field

The equation is $u_t = \varepsilon\Delta u + (u - u^3)/\varepsilon$ with periodic boundary conditions on $[0, 1]^2$. The coarse solver uses second-order central differences for the Laplacian and explicit Euler for the reaction term. The reference solver uses the same scheme on a grid $8\times$ finer. Parameters: $\varepsilon \in [0.002, 0.1]$, $N_c \in \{16, 32\}$, $T = 0.4$. The corrector learns to resolve the interphase front of width ε. Solutions are bounded in $[-1, 1]$.

B Corrector architectures and training

B.1 Full corrector

The full corrector is a plain CNN: 24 channels (2D), 16 channels (3D), or 32 channels (1D), with 3×3 (2D/3D) or 5×5 (1D) circular convolutions, GELU activations, and Adam (lr 10^{-3}, or 2×10^{-3} for 3D). The network has 3 convolutional layers. Full training uses all 256 training pairs, the full epoch budget ($E_{\max}$), no early stopping, and 3 seeds; the reported full-corrector error is the mean across seeds.

B.2 Pilot corrector

The pilot corrector is an 8-channel CNN with the same architecture at reduced width. The pilot uses 26 pairs (10% of 256), 20% of the epoch budget ($E_A = 0.2\,E_{\max}$), and seed 1000. The pilot is trained on the first 26 pairs only; the full corrector is trained on all 256 pairs.

B.3 U-Net variant

A U-Net variant (channel-matched additive outer skip) is used for the architecture-transfer check. The U-Net has the same width as the plain CNN but adds skip connections from the encoder to the decoder.

B.4 Training protocol

All models are trained with Adam (lr 10^{-3}, or 2×10^{-3} for 3D), MSE loss, and no weight decay. The epoch budget $E_{\max}$ is problem-specific; the pilot uses $E_A = 0.2\,E_{\max}$. Seeds are fixed for Python, NumPy, and PyTorch; the pilot uses seed 1000. Because GPU kernels are not bitwise deterministic, we report per-seed and mean errors rather than claiming exact determinism.

C Cost accounting

Decision cost. The decision cost of a selector is the compute it spends to reach a Learn/Refine verdict, as a fraction of the full corrector's end-to-end training cost (reference-solve data generation + optimization). For the pilot-based selectors (raw pilot, B2-d10, M1b) this is the shared 8-channel

pilot corrector: 10% of the training pairs plus 20% of the epoch budget, plus the cheap classical frontier (ridge + refinement arms), which requires no training. All three use the *same* pilot and so have the same decision cost: on the frozen Allen–Cahn set this is a mean of 14.7% of full training (median 14.7%, max 15.6%). M1b additionally reads the pilot's error at 5% and 10% of the budget, but these are recorded along a single training run and add negligible cost. The classical-only probe trains no neural corrector, so its cost is only the shared classical frontier (the ridge screen and refinement arms): a mean of 4.9% of full training on Allen–Cahn, strictly below the pilot-based selectors because it adds no neural pilot. Estimator A extrapolates the learning curve from the full dataset and costs $\approx 79\%$.

Decision-time vs ground-truth classical arm. The ground-truth classical arm e^*_{cls} is the best arm affordable under the *true* full learned cost L^* (Eq. 1). The arm the method uses at decision time is computed under its *estimated* budget, since the true full cost is not known prospectively. The two differ for all 10 Allen–Cahn regimes (the estimate uses a fixed inference constant), but no Learn/Refine decision flips between them, and the regret oracle uses the ground-truth arm e^*_{cls} throughout.

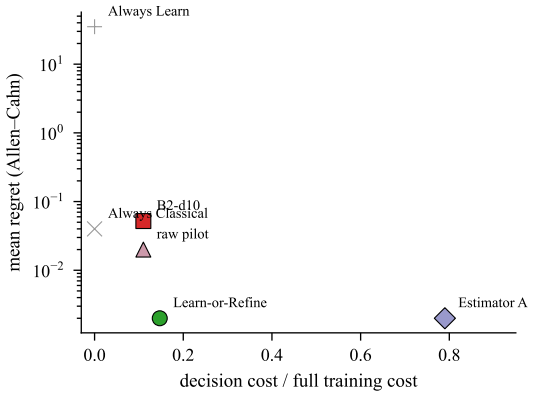

Figure 7: Cost–regret frontier on the Allen–Cahn confirmatory set. Learn-or-Refine sits at the low-cost, zero-regret corner; Estimator A matches regret at $\sim 5\times$ the cost.

D Complete regime tables

Table 4 lists all 74 historical regime×architecture cases used for the leave-one-family-out analysis. Table 5 lists the 10 frozen Allen–Cahn confirmatory regimes with their true and predicted margins.

Table 4: Per-regime results for the 74 historical regime×architecture cases. Errors are relative L_2 against the reference; $g = \log(e_{\text{pilot}}/e_{\text{full}})$; $S^* = \min(\log e_{\text{ridge}}/e_{\text{full}}, \log e_{\text{cls}}/e_{\text{full}})$; winner is the matched-cost ground truth.

case	family	dim	arch	e_{ridge}	e_{cls}	e_{full}	e_{pilot}	g	S^*	winner
D10_3d_nu2e-2_N16_cnn	AD	3	cnn	0.1633	0.0558	0.0073	0.0572	2.05	2.03	learned
D1_2d_nu5e-3_N32_cnn	AD	2	cnn	0.0903	0.0302	0.0159	0.0634	1.39	0.64	learned
D2_3d_nu5e-3_N24_cnn	AD	3	cnn	0.0776	0.0172	0.0196	0.0358	0.60	-0.13	classical
D3_2d_nu2e-3_N32_cnn	AD	2	cnn	0.1276	0.0474	0.0201	0.0537	0.98	0.86	learned
D4_2d_nu1e-3_N16_cnn	AD	2	cnn	0.3116	0.1214	0.0594	0.1637	1.01	0.72	learned
D5_3d_nu1e-3_N16_cnn	AD	3	cnn	0.1217	0.0381	0.0313	0.0401	0.25	0.20	learned
D6_2d_nu1e-2_N32_cnn	AD	2	cnn	0.0836	0.0335	0.0197	0.0476	0.88	0.53	learned
D7_3d_nu1e-2_N12_cnn	AD	3	cnn	0.2079	0.0610	0.0420	0.0776	0.61	0.37	learned
D8_2d_nu5e-2_N32_cnn	AD	2	cnn	0.0044	0.0130	0.0034	0.0235	1.95	0.27	learned
D9_2d_nu2e-2_N16_cnn	AD	2	cnn	0.1350	0.0436	0.0312	0.0514	0.50	0.34	learned

continued

case	family	dim	arch	e_{ridge}	e_{cls}	e_{full}	e_{pilot}	g	S^*	winner
F1_burgers_smooth_cnn	Burg	1	cnn	0.0410	0.0035	0.0256	0.0791	1.13	-1.99	classical
F2_burgers_shock_cnn	Burg	1	cnn	0.4036	0.1288	0.2018	0.3593	0.58	-0.45	classical
F3_advdiff_1d_cnn	AD	1	cnn	0.1059	0.0367	0.0338	0.0642	0.64	0.08	learned
F4_helmholtz_cnn	Helm	1	cnn	0.5735	0.0652	0.5705	0.5027	-0.13	-2.17	classical
H1_2d_nu2e-2_N32_cnn	AD	2	cnn	0.0294	0.0236	0.0074	0.0304	1.41	1.15	learned
H2_2d_nu1e-2_N16_cnn	AD	2	cnn	0.1540	0.0526	0.0405	0.1190	1.08	0.26	learned
H3_3d_nu5e-3_N16_cnn	AD	3	cnn	0.1170	0.0286	0.0342	0.0516	0.41	-0.18	classical
H4_2d_nu2e-3_N16_cnn	AD	2	cnn	0.3116	0.1115	0.0478	0.1415	1.08	0.85	learned
RA_2d_nu1e-3_N32_cnn	AD	2	cnn	0.1515	0.0528	0.0230	0.0634	1.01	0.83	learned
RB_2d_nu5e-3_N16_cnn	AD	2	cnn	0.2327	0.0866	0.0424	0.1365	1.17	0.71	learned
RC_3d_nu2e-3_N24_cnn	AD	3	cnn	0.0780	0.0236	0.0267	0.0351	0.28	-0.12	classical
A01_cnn	AD	1	cnn	0.1114	0.0404	0.0403	0.0589	0.38	0.00	learned
A02_cnn	AD	1	cnn	0.0519	0.0249	0.0468	0.0507	0.08	-0.63	classical
A03_cnn	AD	1	cnn	0.0392	0.0127	0.0393	0.0418	0.06	-1.13	classical
B01_cnn	Burg	1	cnn	0.0255	0.0019	0.0255	0.0288	0.12	-2.62	classical
B02_cnn	Burg	1	cnn	0.1110	0.0080	0.0452	0.0873	0.66	-1.73	classical
B03_cnn	Burg	1	cnn	0.1832	0.0157	0.0847	0.1751	0.73	-1.69	classical
B04_cnn	Burg	1	cnn	0.4292	0.1970	0.2878	0.3500	0.20	-0.38	classical
H01_cnn	Helm	1	cnn	0.9513	0.0867	0.9024	0.9411	0.04	-2.34	classical
H02_cnn	Helm	1	cnn	0.0900	0.0202	0.0888	0.1139	0.25	-1.48	classical
R2D01_cnn	AD	2	cnn	0.1212	0.0487	0.0215	0.0633	1.08	0.82	learned
R2D02_cnn	AD	2	cnn	0.0718	0.0259	0.0131	0.0353	0.99	0.68	learned
R2D03_cnn	AD	2	cnn	0.0511	0.0200	0.0134	0.0259	0.66	0.40	learned
R2D04_cnn	AD	2	cnn	0.1449	0.0455	0.0304	0.0824	1.00	0.40	learned
R2D05_cnn	AD	2	cnn	0.0205	0.0229	0.0070	0.0376	1.69	1.08	learned
R2D06_cnn	AD	2	cnn	0.0948	0.0365	0.0312	0.0767	0.90	0.16	learned
R3D01_cnn	AD	3	cnn	0.0755	0.0236	0.0230	0.0354	0.43	0.03	learned
R3D02_cnn	AD	3	cnn	0.1383	0.0403	0.0394	0.0595	0.41	0.02	learned
R3D03_cnn	AD	3	cnn	0.1192	0.0417	0.0327	0.0345	0.05	0.24	learned
R3D04_cnn	AD	3	cnn	0.2266	0.0677	0.0385	0.0890	0.84	0.56	learned
R3D05_cnn	AD	3	cnn	0.0785	0.0256	0.0236	0.0373	0.46	0.08	learned
B2D01_cnn	Burg	2	cnn	0.1958	0.0222	0.1155	0.1417	0.20	-1.65	classical
B2D01_unet	Burg	2	unet	0.1958	0.0222	0.1152	0.1427	0.21	-1.65	classical
B2D02_cnn	Burg	2	cnn	0.2915	0.0534	0.1558	0.2043	0.27	-1.07	classical
B2D02_unet	Burg	2	unet	0.2915	0.0534	0.1587	0.2110	0.28	-1.09	classical
B2D03_cnn	Burg	2	cnn	0.4137	0.1265	0.2129	0.2958	0.33	-0.52	classical
B2D03_unet	Burg	2	unet	0.4137	0.1265	0.2404	0.3759	0.45	-0.64	classical
B2D04_cnn	Burg	2	cnn	0.5221	0.2527	0.2788	0.3573	0.25	-0.10	classical
B2D04_unet	Burg	2	unet	0.5221	0.2527	0.3216	0.3788	0.16	-0.24	classical
B2D05_cnn	Burg	2	cnn	0.1660	0.0194	0.1688	0.1761	0.04	-2.16	classical
B2D05_unet	Burg	2	unet	0.1660	0.0194	0.1563	0.1549	-0.01	-2.09	classical
B2D06_cnn	Burg	2	cnn	0.4025	0.0914	0.2455	0.2535	0.03	-0.99	classical
B2D06_unet	Burg	2	unet	0.4025	0.0914	0.3230	0.3411	0.05	-1.26	classical
B2D07_cnn	Burg	2	cnn	0.3682	0.0945	0.2048	0.2465	0.19	-0.77	classical
B2D07_unet	Burg	2	unet	0.3682	0.0945	0.2129	0.2575	0.19	-0.81	classical
B2D08_cnn	Burg	2	cnn	0.5551	0.3206	0.3026	0.3647	0.19	0.06	learned
B2D08_unet	Burg	2	unet	0.5551	0.3206	0.3520	0.4033	0.14	-0.09	classical
FK1_cnn	FK	1	cnn	0.0000	0.0000	0.0001	0.0007	2.35	-5.35	classical
FK2D1_cnn	FK	2	cnn	0.0004	0.0000	0.0003	0.0004	0.25	-4.71	classical
FK2D1_unet	FK	2	unet	0.0004	0.0000	0.0005	0.0026	1.68	-5.25	classical
FK2D2_cnn	FK	2	cnn	0.0009	0.0000	0.0005	0.0008	0.51	-3.97	classical
FK2D2_unet	FK	2	unet	0.0009	0.0000	0.0006	0.0036	1.83	-4.16	classical
FK2D3_cnn	FK	2	cnn	0.0003	0.0000	0.0002	0.0003	0.19	-4.79	classical
FK2D3_unet	FK	2	unet	0.0003	0.0000	0.0003	0.0023	1.96	-5.07	classical
FK2D4_cnn	FK	2	cnn	0.0014	0.0000	0.0012	0.0013	0.10	-3.80	classical
FK2D4_unet	FK	2	unet	0.0014	0.0000	0.0013	0.0026	0.68	-3.89	classical

continued

case	family	dim	arch	e_{ridge}	e_{cls}	e_{full}	e_{pilot}	g	S^*	winner
FK2D5_cnn	FK	2	cnn	0.0001	0.0000	0.0001	0.0001	0.13	-5.03	classical
FK2D5_unet	FK	2	unet	0.0001	0.0000	0.0002	0.0015	2.17	-5.34	classical
FK2D6_cnn	FK	2	cnn	0.0010	0.0000	0.0006	0.0010	0.44	-4.73	classical
FK2D6_unet	FK	2	unet	0.0010	0.0000	0.0007	0.0042	1.74	-4.91	classical
FK2_cnn	FK	1	cnn	0.0001	0.0000	0.0001	0.0006	1.88	-4.63	classical
FK3_cnn	FK	1	cnn	0.0008	0.0000	0.0003	0.0013	1.42	-3.43	classical
FK4_cnn	FK	1	cnn	0.0006	0.0000	0.0002	0.0006	1.28	-2.99	classical
FK5_cnn	FK	1	cnn	0.0008	0.0000	0.0008	0.0014	0.52	-3.52	classical

Table 5: Frozen prospective Allen–Cahn evaluation (10 regimes). ε is the phase-field parameter; truth is the matched-cost ground truth; pred is the frozen Learn-or-Refine decision; regret is the decision regret.

regime	ε	e_{full}	e_{pilot}	e_{ridge}	e_{cls}	S^*	$\widehat{S}$	truth	pred	regret
AC02	0.02	0.0436	0.0436	0.1355	0.0004	-4.70	-3.65	classical	classical	0.0000
AC03	0.01	0.1180	0.1732	0.4589	0.0166	-1.96	-1.79	classical	classical	0.0000
AC04	0.005	0.1201	0.2034	0.3565	0.1444	0.18	0.26	learned	learned	0.0000
AC05	0.1	0.0059	0.0079	0.0079	0.0002	-3.65	-3.55	classical	classical	0.0000
AC06	0.05	0.0225	0.0316	0.0515	0.0002	-4.84	-4.58	classical	classical	0.0000
AC10	0.02	0.2231	0.3360	0.4738	0.0229	-2.28	-2.81	classical	classical	0.0000
AC11	0.005	0.3490	0.3266	0.3483	0.3641	-0.00	0.22	classical	learned	0.0000
AC13	0.03	0.1382	0.1558	0.2749	0.0021	-4.18	-4.18	classical	classical	0.0000
AC14	0.008	0.3120	0.3134	0.4558	0.3735	0.18	0.51	learned	learned	0.0000
AC15	0.002	0.0381	0.0450	0.0333	0.0250	-0.42	-0.50	classical	classical	0.0000

E FULL LOFO RESULTS

Figure 8 shows the distribution of the remaining learning gain g by family; the family-level heterogeneity is the reason a single absolute calibration cannot transfer.

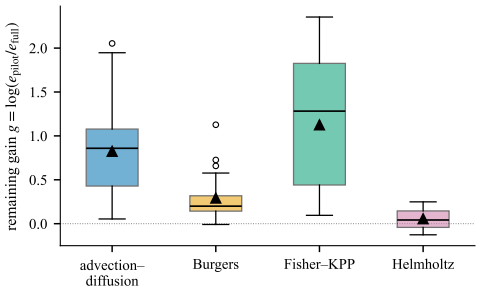

Figure 8: The pilot-to-full remaining gain $g = \log(e_{\text{pilot}}/e_{\text{full}})$ is strongly family-dependent (box: quartiles; triangle: mean).

Per-family LOFO breakdown (micro regret, accuracy) for each estimator:

M1b and B2-d10 are tied on Fisher–KPP and Helmholtz (zero regret) and on advection–diffusion (both 0.132, driven by the single D8 failure). M1b's advantage is concentrated on Burgers, where the frozen absolute calibration over-estimates the gain and produces confident false positives (0.042 vs M1b's 0.022).

F PROSPECTIVE ALLEN–CAHN DETAILS

The Allen–Cahn regimes span $\varepsilon \in \{0.002, 0.005, 0.008, 0.01, 0.02, 0.03, 0.05, 0.1\}$ at $N_c \in \{16, 32\}$ with $T = 0.4$. Ground truths: 2 Learn (AC04, AC14; sharp fronts), 8 Refine. M1b cor-

Table 6: LOFO cross-family regret by family (per-family-mean regret; the macro aggregate of Table 1 decomposed). Each column holds out one family.

Method	AD	Burgers	Fisher–KPP	Helmholtz
Always Learn	0.108	2.949	104.82	6.850
Always Classical	0.863	0.003	0.000	0.000
raw pilot	0.855	0.003	0.000	0.000
B2-d10 (frozen)	0.132	0.042	0.000	0.000
M1b	**0.132**	**0.022**	**0.000**	**0.000**

rectly identifies both Learn cases and 7 of 8 Refine cases. The single miss (AC11) is a zero-margin regime ($e_{\text{full}} \approx e_{\text{cls}} \approx e_{\text{ridge}}$) where any rule flips a coin; the regret is zero.

G Numerical verification

Each solver was verified before any learning. Convergence orders: advection–diffusion $p \approx 1$ (upwind), Burgers $p \approx 1$ (upwind), Fisher–KPP $p \approx 2$ (central), Allen–Cahn $p \approx 2.2$–2.6 (central). Reference-convergence gaps confirmed the coarse→reference hierarchy is adequate. Fisher–KPP and Allen–Cahn solutions were verified to remain bounded in $[0, 1]$ and $[-1, 1]$ respectively.

H Cost breakdown

The training-data cost dominates the learned corrector's end-to-end cost. Per-rollout costs and upfront costs for three representative regimes:

Table 7: Cost breakdown (seconds). Per-rollout cost is what the standard protocol reports; the learned corrector's training-data cost is spent upfront and is two to four orders of magnitude larger than one fine solve.

	1D Burgers smooth	2D advection–diffusion	2D Navier–Stokes
coarse solve /rollout	0.0073	0.0048	0.0098
fine solve ($h/2$) /rollout	0.0366	0.0213	0.0494
finer solve ($h/4$) /rollout	0.1333	0.1188	0.2947
learned corrector /rollout	0.0078	0.0054	0.0103
learned training-data	286	479	403
learned training	5.3	34.7	12

I Statistical significance of the canonical win

The decision-critical comparisons are statistically solid. On 2D Darcy, with $n_{\text{test}} = 100$ episodes and five seeds, the best classical arm ($S_{h/4}$) beats the CNN corrector in $100/100$ episodes (paired win fraction 1.000, bootstrap CI $[1.000, 1.000]$; sign test $p = 1.6 \times 10^{-30}$), an $11.6\times$ error gap that holds for every seed. On 2D advection–diffusion, the learned corrector wins $70/100$ episodes (bootstrap CI $[0.61, 0.79]$, excluding 0.5; sign test $p = 7.9 \times 10^{-5}$), stable across all five seeds (per-seed win fractions 0.65–0.71). The canonical learned win is thus statistically significant but narrow: a $70/30$ majority, not a blowout.

J Robustness to corrector capacity and data

We sweep corrector capacity (16, 32, 64 channels) and training-data budget (256, 1024, 2048 high-fidelity solves) on 1D smooth Burgers. The learned corrector's error plateaus at 0.037–0.051 across the entire sweep, while Richardson($h/2, h/4$) attains 0.0040. More capacity does not help, and quadrupling the training data reduces the learned error only from 0.050 to 0.037, still $9\times$ worse than the classical arm. The 1D gap is not an artifact of an undertrained corrector.

On the winning 2D advection–diffusion regime the learned win is capacity-conditional: at 16 channels the classical arm wins (ratio 0.75), at 32 the win is marginal (1.01), and only at 64 is it clear (1.18); the data budget has little effect. The 3D classical win is unchanged by the same test: classical wins at every capacity and data budget (ratios 0.83–0.90).

K Architecture independence

A natural objection is that our small CNN is the wrong corrector. We re-ran the two decision-critical regimes with a Fourier Neural Operator corrector (Li et al., 2020) (4 spectral layers, width 32, 12 modes, held-out-validation early stopping). The FNO corrector does not change either verdict. On 2D Darcy it attains 0.0042, far better than our CNN, but the fine classical solve attains 0.0007, so classical refinement still wins by $6.0\times$. On 2D advection–diffusion it attains 0.0176, beating the best classical arm by $1.84\times$, a larger margin than our CNN's $1.19\times$. A stronger corrector loses harder in the regimes the boundary classifies as classical wins and wins harder in those it classifies as learned wins.

L Published training-free corrector

We run the published PhysicsCorrect corrector (Huang & Perdikaris, 2026) directly using the authors' public code. In our discretization, the correction is a training-free Newton step on the discrete PDE residual. On 2D Darcy it returns the coarse solution unchanged, with error 0.0531 identical to the uncorrected coarse solve: for the linear elliptic residual its single Newton step is the exact solve of $A_h u = f$, which is classical defect correction and a no-op on an already-converged coarse solve. Grid refinement attains 0.0027, twenty times better. On 2D advection–diffusion the same published corrector is a no-op (0.457, equal to the coarse solve) under the faithful explicit residual, and worse than the coarse solve (0.518) under its implicit-residual variant; both are far above our CNN corrector (0.028) and the best classical arm (0.033). The published training-free corrector changes neither verdict, and on elliptic problems it is classical defect correction by construction.

M A checklist for learned-corrector evaluation

Our audit reduces to a short list of controls that any learned-corrector paper should report; each is cheap and each addresses a way the standard protocol can mislead.

1. **Classical refinement baseline:** report the error of the solver at $h/2$ and $h/4$ on the same test set and rollout count.
2. **Richardson extrapolation baseline:** report $R_{h,h/2}$ and $R_{h/2,h/4}$ with the empirically-estimated order p and the amplification factor $(2^p + 1)/(2^p - 1)$.
3. **A low-parameter learned control:** fit a few-parameter map (e.g. ridge on local stencil features) to the same residual; if it matches the neural corrector, capacity is not the source of the gain.
4. **End-to-end cost:** report training-data generation and training cost, not just inference cost, and state the rollout volume.
5. **Breakeven volume:** state the rollout volume N^* at which the learned corrector breaks even against the best affordable classical arm.

N Deployment-volume sensitivity

The matched-cost decision depends on N through equation 1. The corrector's up-front cost (n_{train} reference solves + training) is fixed, so the per-rollout budget C_{learn}/N shrinks as N grows. At small N the budget per rollout is large, so the expensive classical arms ($h/4$, Richardson) are affordable and classical refinement can match the corrector; at large N the budget per rollout approaches the corrector's cheap inference cost, so only cheap classical arms remain affordable and the corrector wins more often. Recomputing the decisions across the historical regimes for

$N \in \{10, 100, 10^3, 10^4, 10^5\}$ gives a monotone flip pattern: several regimes move from Refine at $N = 10$ to Learn at $N \geq 10^4$, and no regime moves from Learn to Refine. This is the expected behavior of a cost-aware selector, not instability; the $N = 1000$ value used throughout is a pre-registered convention, and the deployment volume is a parameter of the decision, not of the method.

O FAILURE ANALYSIS

The catastrophic historical failure is regime `D8_2d_nu5e-2_N32` (2D advection–diffusion, $\nu = 0.05$, $N_c = 32$, CNN). The full corrector reaches $e_{\text{full}} = 0.0034$, far below the classical arm $e_{\text{cls}} = 0.0130$ and the ridge $e_{\text{ridge}} = 0.0044$, so the true margin is positive ($S^* = 0.27$, winner = learned). The pilot, however, has nearly plateaued ($p_{\text{early}} = 0.040$, $p_{\text{late}} = 0.086$), so the frozen estimator predicts a small remaining gain $\widehat{g} \approx 0.36$, whereas the true gain is $g = 1.95$. The predicted full error $\widehat{e}_{\text{full}} \approx 0.017$ therefore sits above the classical threshold and the method predicts Refine — a false negative with regret $(e_{\text{cls}} - e_{\text{full}})/e_{\text{full}} = 2.88$, the largest in the dataset. The frozen absolute calibration (B2-d10) fails identically. The mechanism is transparent: when the achievable error is far below what the pilot can reach, the pilot's early trajectory carries no signal of it, so the remaining gain is under-estimated precisely when the corrector is unusually strong. This is a single hard regime, not a family-wide failure.

P PROTOCOL AMENDMENTS

The Allen–Cahn confirmatory set was subject to two pre-outcome computational amendments, made before any of the affected regimes' outcomes were observed.

Amendment B (regime substitution). Three fine-grid regimes ($N_c = 48$; AC07, AC08, AC12) were computationally infeasible: the Allen–Cahn diffusion CFL ($\Delta t \propto \Delta x^2/\varepsilon$) makes the $N_R = 8N_c = 384$ reference solve take roughly 2–3 hours per regime. They were replaced by three fast $N_c = 16$ regimes spanning the same ε range (AC13, AC14, AC15), preserving ε coverage and the coarse/well-resolved spread.

Amendment C (further drops). Two large-ε, $N_c = 32$ regimes (AC01, AC09) were dropped: the same CFL makes their reference solves take 1.5–5 hours under CPU contention. Both are smooth and well-resolved (a-priori classical). The final set of 10 regimes spans $\varepsilon \in [0.002, 0.1]$ at $N_c \in \{16, 32\}$, with both sharp-front and smooth cases.