ArtificialGate

ArtificialGate Ltd. Research Preprint

The Direction-Execution Gap in Physics-Informed Neural Network Optimization

Barak Or
ArtificialGate Ltd., Israel
barak@artificialgate.ai

Abstract

PINN optimization is often interpreted through component-gradient geometry, but training acts through a finite, optimizer-state-dependent displacement. We introduce the *direction-execution gap*: a component predicts local descent along the displacement that is ultimately executed, yet its loss increases after that displacement is completed. The proposed fixed-set audit evaluates the actual optimizer map on identical points before and after execution, isolating parameter-induced change from collocation replacement. In 111 paired runs across four PDE families and three update rules, first-order sign accuracy was 86.39% and reversal prevalence 13.56%. One directional Hessian-vector product raised sign accuracy to 99.63% and reduced the median normalized magnitude-error ratio to 1.284×10^{-3}, a 99.87% reduction. These results establish a directly measurable reliability object for the complete optimizer map and show that local directional promise can diverge from realized component progress at practical PINN step lengths. A 30-run curvature-aware acceptance study reduced median reversal prevalence from 11.61% under Adam to 3.44%, while final-error comparisons remained statistically indistinguishable from matched controls. This establishes executed-step reliability as an actionable objective complementary to solver accuracy.

Keywords. physics-informed neural networks; finite-step reliability; adaptive optimization; directional curvature; multi-objective learning.

1 The Direction-Execution Gap

Physics-informed neural networks (PINNs) approximate solutions of partial differential equations (PDEs) by jointly optimizing interior residual, boundary, initial, or observational losses [1]. Training failures have been linked to operator stiffness, gradient imbalance, ill-conditioned loss landscapes, and unreliable solver-level accuracy [2-7]. Because the component terms define a multi-objective problem, related responses include adaptive PINN loss balancing and Pareto- or projection-based gradient aggregation [8-10]. A recent review further organizes PINN optimization around parameter search, architecture, sampling, generalization, and evolutionary alternatives [29].

PDEs with closed-form or manufactured solutions remain useful PINN benchmarks because they provide exact error references and isolate optimization effects under controlled physics. The practical motivation is broader: PINNs combine differential constraints with sparse observations in inverse, data-assimilation, and geometrically complex settings where an analytic solution is unavailable. The canonical equations used here therefore test the optimizer mechanism before it is transferred to those harder regimes.

DCGD and ConFIG reconcile component gradients, while other work studies second-order alignment, loss-landscape curvature, residual-surface flattening, and metric-aware or quasi-Newton updates [5, 11-16]. PCGrad also connects gradient conflict to aggregate multi-task curvature across successive iterates [10]. These methods analyze or construct directions, weights, or global step controls. They do not directly test whether a component that is locally favorable along

This research was conducted under ArtificialGate Ltd. The author serves as Academic Director at the Google and Reichman Tech School, and as an External Lecturer at Technion-Israel Institute of Technology and Reichman University.

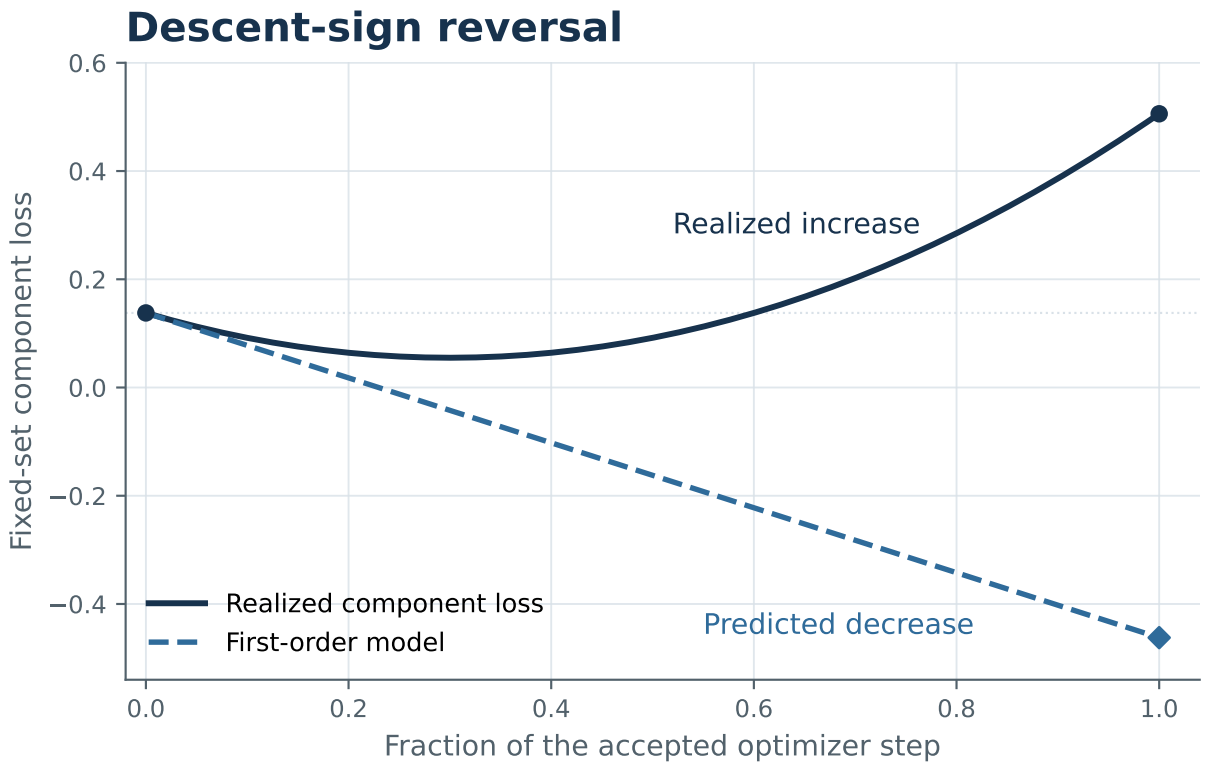

Figure 1: Direction-execution gap: a locally favorable first-order prediction can end at a higher loss after the accepted finite optimizer step.

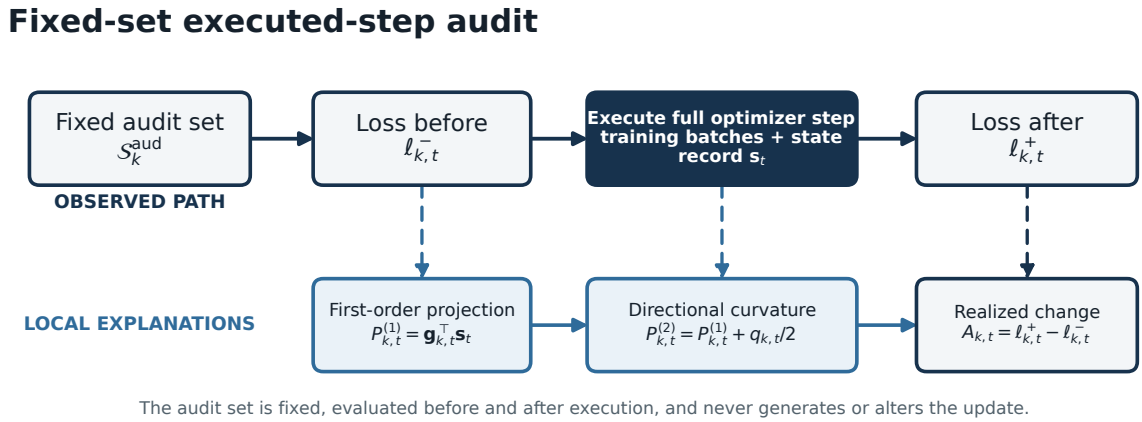

Figure 2: Fixed-set executed-step audit. The upper path measures loss before and after the complete optimizer step; the lower path explains the realized change using the recorded displacement. The fixed audit set never generates or alters the update.

the final recorded displacement actually improves after the stateful optimizer map is executed on unchanged points. The open question is therefore not whether curvature exists, but whether the complete finite step realizes the component-wise local claim.

We call this distinction the *direction-execution gap*. If a component derivative has a favorable projection on the recorded update but the same component loss increases on the same points, a *descent-sign reversal* has occurred. Figures 1 and 2 separate the event from the measurement needed to distinguish it from collocation resampling.

The phenomenon can occur in any differentiable multi-loss network, but PINNs make it especially consequential: their components encode distinct physical constraints, collocation resampling can confound before/after comparisons, and lower residual loss does not necessarily imply lower solution error.

This work contributes: (i) a component-wise reliability object for the complete executed optimizer map; (ii) a paired fixed-set audit that measures it without collocation replacement and has no larger variance under nonnegative pairing covariance; and (iii) systematic evidence that local directional promise and realized progress diverge at practical PINN step lengths. The novelty lies in the measured object, the isolation protocol, and the empirical reliability analysis; the Hessian-vector product explains the mechanism. A controlled intervention verifies that the reliability target can be reduced directly.

The remainder of this paper develops the audit in Section 2, specifies the experiments in Section 3, reports the results

in Section 4, discusses scope in Section 5, and provides derivations and statistical details in Appendix A.

2 A Fixed-Set Executed-Step Audit

2.1 Component Losses and Update-Generating Batches

Let $u_{\boldsymbol{\theta}} : \Omega \subset \mathbb{R}^{d_z} \to \mathbb{R}^{d_u}$ be a PINN with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$. Here d_z is the input dimension, including spatial and, when present, temporal coordinates; d_u is the number of predicted physical fields; and p is the number of trainable parameters. For component $k \in \{1, \dots, K\}$, the operator $\mathcal{R}_k$ evaluates a PDE, boundary, or initial-condition residual with target b_k. On a finite sample set $\mathcal{S}$, the raw component loss is

$$\mathcal{L}_k(\boldsymbol{\theta}; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{z \in \mathcal{S}} \|\mathcal{R}_k[u_{\boldsymbol{\theta}}](z) - b_k(z)\|_2^2. \tag{1}$$

A sample item z can be a single coordinate or a tuple of coordinates required by a coupled boundary operator.

At step t, the training objective supplied to the optimizer is

$$\mathcal{L}_{\mathrm{tr},t}(\boldsymbol{\theta}) = \sum_{k=1}^{K} w_{k,t} \mathcal{L}_k(\boldsymbol{\theta}; \mathcal{B}_{k,t}), \tag{2}$$

where tr denotes training, $\mathcal{B}_{k,t}$ is the component-k batch that generates the update, and $w_{k,t} > 0$ is its weight. The training batches may vary with t, whereas the audit sets remain fixed throughout the run.

2.2 Frozen Audit Losses and the Executed Step

Each component is assigned a fixed audit set $\mathcal{S}_k^{\mathrm{aud}}$. It is drawn once per run, is disjoint from the independent solution-evaluation set, and is never used to calculate the training update. The identical audit points are evaluated before and after each audited update, so the measured difference reflects parameter change rather than collocation replacement. The losses are

$$\ell_{k,t}^{-} = \mathcal{L}_k(\boldsymbol{\theta}_t; \mathcal{S}_k^{\mathrm{aud}}), \tag{3}$$

$$\ell_{k,t}^{+} = \mathcal{L}_k(\boldsymbol{\theta}_{t+1}; \mathcal{S}_k^{\mathrm{aud}}). \tag{4}$$

The executed parameter displacement is

$$\boldsymbol{s}_t = \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t, \tag{5}$$

including the complete effect of momentum, adaptive preconditioning, or conflict resolution. The realized component-loss change is

$$A_{k,t} = \ell_{k,t}^{+} - \ell_{k,t}^{-}. \tag{6}$$

Thus $A_{k,t} < 0$ denotes improvement on the audit set, whereas $A_{k,t} > 0$ denotes deterioration.

2.3 Why the Audit Points Are Held Fixed

Proposition 1 (fixed-set isolation) Let Z follow the component-k audit design, let $f_k^{-}(Z)$ and $f_k^{+}(Z)$ be the pointwise losses at $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+1}$, and let $\bar{f}_k^{\pm}$ denote their sample means. The paired estimator uses one set $\mathcal{S}$, whereas the independent estimator uses independent sets $\mathcal{S}^{+}$ and $\mathcal{S}^{-}$; both are unbiased. Writing $C_{k,t} = \mathrm{Cov}(\bar{f}_k^{+}(\mathcal{S}), \bar{f}_k^{-}(\mathcal{S}))$ gives

$$\mathrm{Var}(\widehat{\Delta}_{\mathrm{pair}}) = \mathrm{Var}(\widehat{\Delta}_{\mathrm{ind}}) - 2C_{k,t}. \tag{7}$$

The result follows by expanding the variance of the paired difference; Appendix A gives the iid specialization. When the paired losses are nonnegatively correlated, pairing has no larger variance. Once the points are frozen, Eqs. (3) to (6) measure one fixed empirical objective exactly, so a sign change cannot be produced by replacing collocation points.

2.4 From Local Direction to Executed Step

At the pre-update parameters, define

$$\boldsymbol{g}_{k,t} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta}_t; \mathcal{S}_k^{\mathrm{aud}}), \qquad \boldsymbol{H}_{k,t} = \nabla_{\boldsymbol{\theta}}^2 \mathcal{L}_k(\boldsymbol{\theta}_t; \mathcal{S}_k^{\mathrm{aud}}). \tag{8}$$

The first-order prediction for the change caused by the executed step is

$$P_{k,t}^{(1)} = \boldsymbol{g}_{k,t}^{\top} \boldsymbol{s}_t, \tag{9}$$

Thus $P^{(1)}_{k,t} < 0$ predicts descent. To capture finite-step curvature, define

$$q_{k,t} = \boldsymbol{s}_t^\top \boldsymbol{H}_{k,t} \boldsymbol{s}_t, \tag{10}$$

$$P^{(2)}_{k,t} = P^{(1)}_{k,t} + \frac{1}{2} q_{k,t}. \tag{11}$$

The product $\boldsymbol{H}_{k,t}\boldsymbol{s}_t$ is computed by differentiating the scalar $\boldsymbol{g}_{k,t}^\top \boldsymbol{s}_t$ with respect to $\boldsymbol{\theta}$, treating the recorded step vector as constant [18, 19]. This Hessian-vector product uses vectors with p entries and never forms a dense $p \times p$ Hessian.

For $\alpha \in [0, 1]$, let $\boldsymbol{\theta}_t(\alpha) = \boldsymbol{\theta}_t + \alpha \boldsymbol{s}_t$ denote the point at fraction α of the executed segment. Define the path Hessian $\boldsymbol{H}_{k,t}(\alpha) = \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_k(\boldsymbol{\theta}_t(\alpha); \mathcal{S}_k^{\text{aud}})$ and the directional curvature $h_{k,t}(\alpha) = \boldsymbol{s}_t^\top \boldsymbol{H}_{k,t}(\alpha) \boldsymbol{s}_t$. Taylor's theorem with integral remainder [17] gives

$$A_{k,t} = P^{(1)}_{k,t} + \int_0^1 (1 - \alpha) h_{k,t}(\alpha)\, d\alpha. \tag{12}$$

Thus first-order error is accumulated directional curvature along the accepted segment; Appendix A gives the derivation.

Proposition 2 (executed-step sign certificate) Assume $\|\boldsymbol{H}_{k,t}(\alpha) - \boldsymbol{H}_{k,t}(0)\|_2 \le \rho_{k,t} \alpha \|\boldsymbol{s}_t\|_2$ for $\alpha \in [0, 1]$. Then

$$\left| A_{k,t} - P^{(2)}_{k,t} \right| \le \frac{\rho_{k,t}}{6} \|\boldsymbol{s}_t\|_2^3 \triangleq r_{k,t}. \tag{13}$$

Appendix A derives the bound and verifies the exact factor $1/6$. This bound is not used to compute the reported empirical metrics because $\rho_{k,t}$ is not estimated; the experiments use the observed change, $P^{(1)}_{k,t}$, $P^{(2)}_{k,t}$, and $\chi_{k,t}$. Its role is to provide an optional sufficient sign certificate. Define the two-sided certificate margins

$$M^{\downarrow}_{k,t} = -(P^{(2)}_{k,t} + r_{k,t}), \qquad M^{\uparrow}_{k,t} = P^{(2)}_{k,t} - r_{k,t}. \tag{14}$$

A positive $M^{\downarrow}_{k,t}$ certifies realized descent, while a positive $M^{\uparrow}_{k,t}$ certifies realized increase; otherwise the local certificate is unresolved. Equivalently, $|P^{(2)}_{k,t}| > r_{k,t}$ certifies its sign, while $|P^{(1)}_{k,t}| > |q_{k,t}|/2 + r_{k,t}$ certifies the first-order sign.

For applications requiring the certificate, directional probes can estimate path variation; Appendix A gives the estimator. A finite probe grid is not a certified spectral-norm upper bound, so rigorous certification requires additional structural or global smoothness information.

Corollary (quadratic reversal threshold) In the locally quadratic case $r_{k,t} = 0$, a predicted descent with $P^{(1)}_{k,t} < 0$ and $q_{k,t} > 0$ is reversed exactly when $q_{k,t}/(2|P^{(1)}_{k,t}|) > 1$. Thus the audit, certificate margin, and curvature ratio answer different questions: what happened, what is guaranteed locally, and why the first-order sign can fail. No global smoothness constant is required for the empirical audit itself.

2.5 Audit Measures

To compare components with different scales, let $c_{k,t} = |\ell^-_{k,t}| + \delta$ and define $\widehat{A}_{k,t} = A_{k,t}/c_{k,t}$ and $\widehat{P}^{(j)}_{k,t} = P^{(j)}_{k,t}/c_{k,t}$. The floor $\delta > 0$ and a sign tolerance $\varepsilon > 0$ exclude roundoff-level changes.

A first-order descent-sign reversal is

$$\widehat{P}^{(1)}_{k,t} < -\varepsilon, \qquad \widehat{A}_{k,t} > \varepsilon. \tag{15}$$

We distinguish two frequencies. *Prevalence* is the fraction of all audited component events that satisfy Eq. (15). *Conditional risk* uses only events for which first order predicted a decrease in the denominator. The distinction matters because an optimizer can predict descent more or less often without changing the reliability of those predictions.

For model $j \in \{1, 2\}$, let $\mathcal{D}^{(j)}$ contain events for which both $|\widehat{A}_{k,t}| > \varepsilon$ and $|\widehat{P}^{(j)}_{k,t}| > \varepsilon$. The run-level sign-prediction accuracy is

$$\text{Acc}^{(j)} = \frac{1}{|\mathcal{D}^{(j)}|} \sum_{(k,t) \in \mathcal{D}^{(j)}} \mathbb{I}\left[\text{sgn}(\widehat{P}^{(j)}_{k,t}) = \text{sgn}(\widehat{A}_{k,t}) \right]. \tag{16}$$

The normalized magnitude error and curvature-to-linear ratio are

$$E^{(j)}_{k,t} = |\widehat{P}^{(j)}_{k,t} - \widehat{A}_{k,t}|, \tag{17}$$

$$\chi_{k,t} = \frac{|q_{k,t}|/2}{|P^{(1)}_{k,t}| + \delta}. \tag{18}$$

Values $\chi_{k,t} \ll 1$ indicate linear dominance; values near or above one indicate that the quadratic correction can match or overturn the linear term. Because $\chi_{k,t}$ is constructed from the same local terms as Eq. (11), it is an explanatory finite-step scale rather than an independently calibrated failure probability.

Algorithm 1 Optimizer-aware audit of an accepted PINN update

Require: $\boldsymbol{\theta}_t$; batches $\{\mathcal{B}_{k,t}\}$; fixed audit sets $\{\mathcal{S}_k^{\text{aud}}\}$; optimizer state
1: Record $\ell_{k,t}^-$ on each audit set using Eq. (3).
2: Execute one complete optimizer step and record $\boldsymbol{s}_t$ using Eq. (5).
3: Re-evaluate the same sets to obtain $A_{k,t}$ using Eqs. (4)-(6).
4: Restore $\boldsymbol{\theta}_t$ temporarily; compute $P_{k,t}^{(1)}$ and the Hessian-vector product using Eqs. (9)-(11).
5: Classify the sign outcomes using Eqs. (15)-(16).
6: Restore $\boldsymbol{\theta}_{t+1}$ and continue training.

Table 1: PDE settings and sample counts. Counts are interior/boundary/initial; "n/a" indicates no initial-condition term.

PDE	Two settings	Train	Audit	Eval.
Poisson 2-D	$(m,n) = (1,1), (4,5)$	768/256/n/a	384/128/n/a	16k
Burgers	$\nu = 0.05/\pi, 0.01/\pi$	768/160/160	384/80/80	16k
Allen-Cahn	$\epsilon = 10^{-3}, 10^{-4}$	768/160/160	384/80/80	16k
Helmholtz	$(m,n) = (1,2), (5,7)$, $\kappa = 10$	1024/256/n/a	512/128/n/a	16k

2.6 Audit Procedure and Cost

Unlike supervisory training schemes that use secondary probes to accept or roll back optimizer proposals [25], Algorithm 1 observes an accepted update but does not modify training. If an ordinary training step costs C_{tr}, one audit checkpoint adds before-and-after component evaluations, one gradient calculation per component, and one Hessian-vector product per component. For K components and D audit checkpoints over T training steps, the cost is approximately

$$O\big(TC_{\text{tr}} + DK(C_{\text{eval}} + C_{\text{grad}} + C_{\text{HVP}})\big)\,. \tag{19}$$

Additional storage is $O(p)$ rather than $O(p^2)$. In the primary runs, $D = 80$ and $T = 6000$, so the audit sampled only 1.33% of optimizer steps.

3 Experimental Protocol

The primary grid spans Poisson 2-D, Burgers, Allen-Cahn, and Helmholtz equations, each with two predefined settings that vary frequency, viscosity, or diffusivity. For Helmholtz, the wave number is set to $\kappa = 10$. Burgers and Allen-Cahn are evaluated against independent numerical reference solvers whose coarse-to-fine differences remain below 1%; Poisson and Helmholtz use manufactured solutions. The 1% value is a conservative design tolerance chosen so that reference-grid uncertainty is small relative to the solver differences being interpreted; it is not proposed as a universal PINN threshold and can be tightened when the application requires it.

All runs use a five-hidden-layer multilayer perceptron with 64 tanh units per layer and Xavier-normal initialization [20]. Inputs are scaled coordinatewise to $[-1, 1]$. Scrambled Sobol sequences generate deterministic, paired low-discrepancy streams for training, audit, and solution-evaluation sets [21, 22]. The architecture is fixed to isolate update geometry. Within every PDE-setting-seed block, methods share initialization, sampling streams, audit sets, and evaluation points. Using one architecture is an experimental control, not a claim of architecture invariance; all conclusions are restricted to the predefined grid.

The primary grid uses Adam, plain SGD, and ConFIG followed by Adam. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon_A = 10^{-8}$ [23]; SGD uses zero momentum [24]. Raw component weights in Eq. (2) are one for Adam and SGD. ConFIG constructs a conflict-aware direction from the same unweighted component gradients before Adam applies its adaptive transform. Learning rates are declared per PDE and range from 5×10^{-5} to 10^{-3}. Each run performs 6000 FP64 updates with 80 audit checkpoints and no gradient clipping.

The controls isolate mechanisms: Adam tests stateful adaptive execution; zero-momentum SGD, unpreconditioned steps; ConFIG+Adam, conflict resolution followed by adaptive transformation; and half-rate Adam, simple shrinkage. A prespecified sweep adds 45 matched AdamW runs on hard Allen-Cahn, Burgers, and Helmholtz at weight decays 10^{-6}, 10^{-4}, and 10^{-2}. Together with the curvature-aware acceptance control, the study covers adaptive state, unpreconditioned steps, conflict-aware direction construction, step shrinkage, decoupled weight decay, and proposal acceptance. Because the audit consumes the recorded displacement rather than optimizer internals, its definitions also apply unchanged to L-BFGS, Adam-to-L-BFGS schedules, adaptive loss weighting, and gradient normalization. Their empirical gap frequencies are not inferred from the present grid and remain a targeted extension.

For FP64 measurements, $\delta = 10^{-14}$ and $\varepsilon = 10^{-9}$. Statistical comparisons use the run, not the checkpoint or component event, as the inferential unit. First- and second-order outcomes are paired by PDE, setting, optimizer, and seed and assessed with the Wilcoxon signed-rank test [26]; this nonparametric test asks whether the paired run-level differences are centered at zero without assuming Gaussian differences. Holm's procedure controls family-wise error over the primary endpoints [27]. The estimand is the median paired change over this predefined experimental grid. The resulting p-values are not interpreted as population-level evidence over unseen PDE families or architectures.

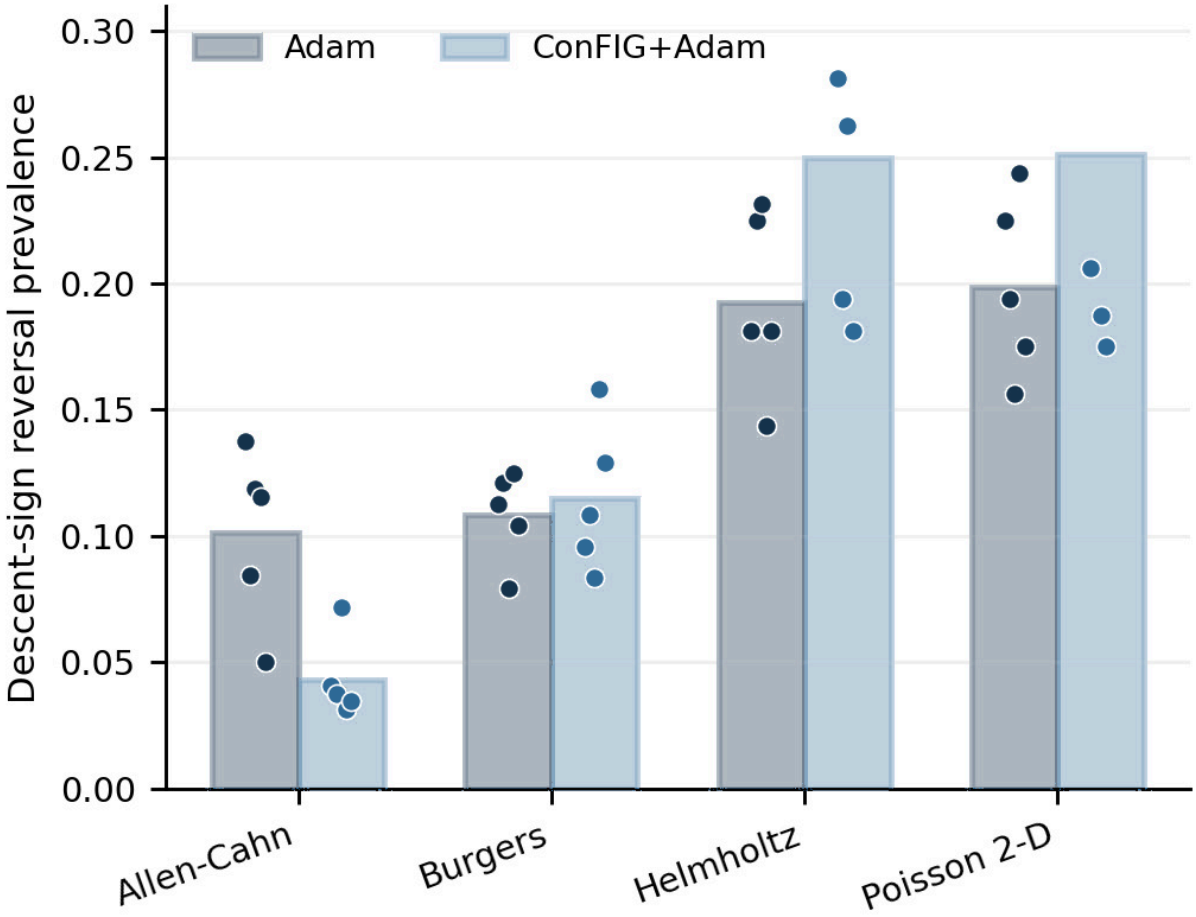

Figure 3: Direction-execution gap in the challenging settings, measured by first-order descent-sign reversal prevalence. Bars show means; points show paired seeds.

4 Measuring the Gap

The primary analysis contains 111 completed runs and 24,960 fixed-set component events. Across the primary grid and auxiliary controls, 186 runs completed in approximately 6.0 aggregate GPU-hours.

For the results below, prevalence divides reversals by all audited events, whereas conditional risk divides reversals only by events for which first order predicted descent. Sign accuracy uses events whose realized and predicted normalized changes exceed the numerical tolerance, and normalized magnitude error is the absolute difference in Eq. (17).

4.1 The Gap Persists Across PDEs and Update Rules

Figure 3 shows that the gap is neither isolated to one PDE nor absent under ConFIG+Adam. Under Adam, reversals accounted for 10.1% of audited component events in Allen-Cahn, 10.8% in Burgers, 19.3% in Helmholtz, and 19.9% in Poisson 2-D for the challenging settings. ConFIG+Adam produced corresponding rates of 4.3%, 11.5%, 25.0%, and 25.1%. This does not contradict ConFIG's favorable projections for its constructed training-batch direction: Adam subsequently transforms that direction, and the audit evaluates the resulting finite step on frozen objectives. The distinction is between direction construction and executed-step progress.

Across all completed primary runs, including both settings and stable SGD runs, first-order sign-prediction accuracy averaged 86.39%. Pooled reversal prevalence was 13.56%, and conditional risk among predicted decreases was 17.85%. Plain SGD generally made smaller updates and exhibited fewer reversals. The pattern is consistent with the finite-step interpretation: a favorable directional derivative guarantees descent only within a sufficiently small neighborhood.

4.2 Directional Curvature Explains the Gap

The curvature-corrected prediction increases mean run-level sign accuracy from 86.39% to 99.63% (Fig. 4). The median within-run gain is 11.88 percentage points. Accuracy improves in 86 runs, is unchanged in 25, and declines in none. The paired run-level comparison remains significant after multiplicity correction ($p = 7.96 \times 10^{-16}$). Second-order reversal prevalence falls to 0.128%, with conditional risk 0.245%. These outcomes span all 23 nonempty PDE, setting, and method cells; they are not produced by one PDE or optimizer stratum.

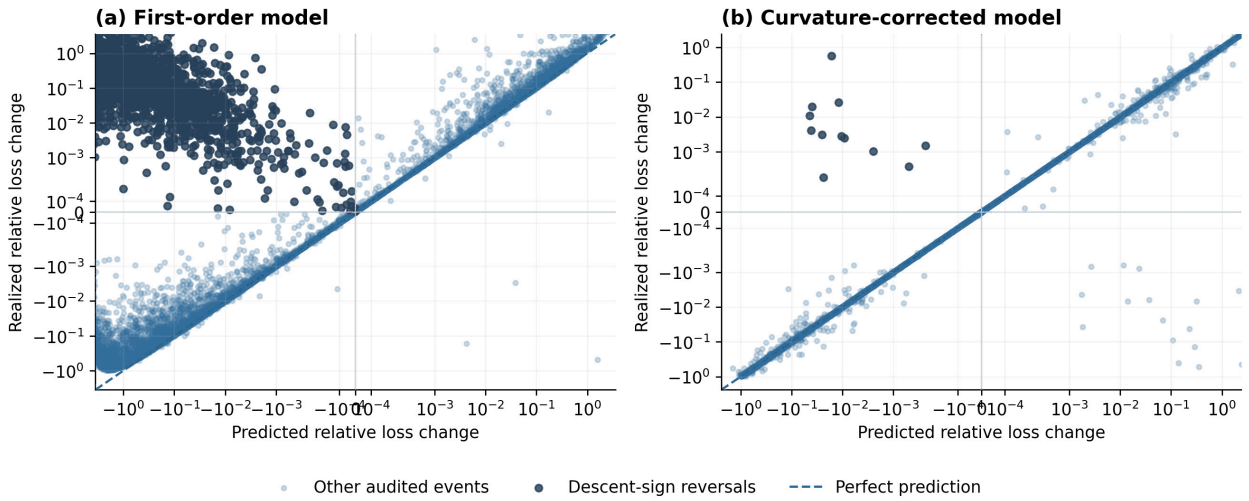

Figure 4: Predicted versus realized fixed-set changes. First order exposes the direction-execution gap; adding local directional curvature concentrates events near the identity line.

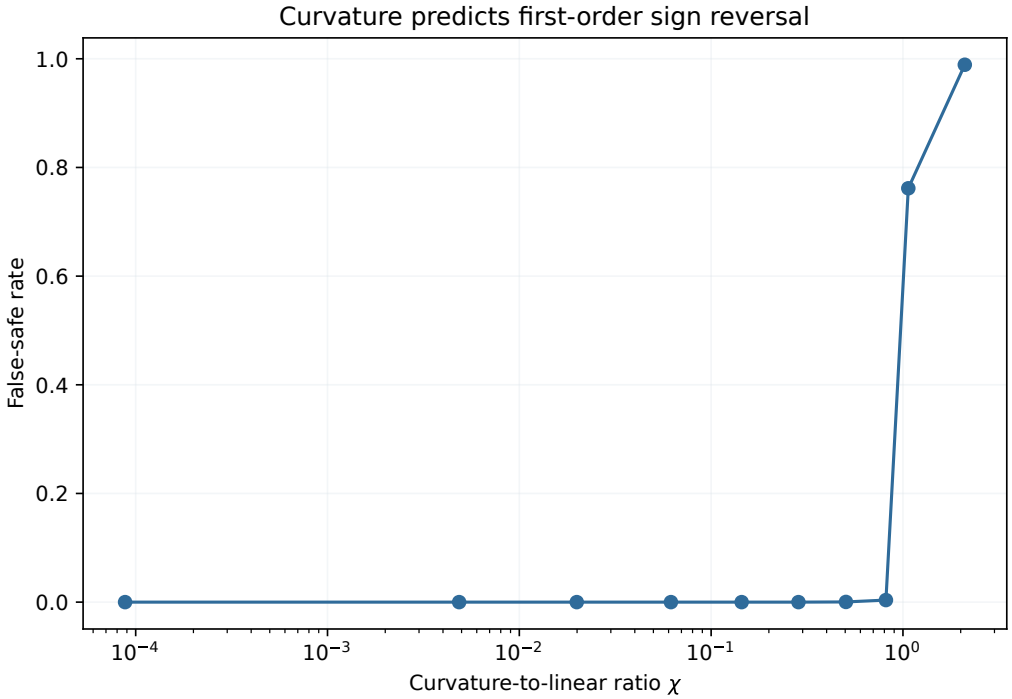

Figure 5: False-safe rate versus the curvature-to-linear ratio. Risk rises when the quadratic contribution becomes comparable with the linear term.

Magnitude recovery is still more consistent. Taking the first-order normalized magnitude error as 1.0, the median second-to-first-order error ratio is 1.284×10^{-3}; equivalently, curvature correction removes 99.87% of that baseline error. Every one of the 111 completed primary runs has lower median magnitude error under the curvature-corrected model ($p = 5.99 \times 10^{-20}$). Figure 5 shows why: reversal frequency rises sharply when the quadratic contribution becomes comparable with the linear term, precisely where the second-order correction can cancel or overturn the first-order sign.

4.3 A Controlled Intervention Validates the Reliability Target

To test whether the diagnostic identifies a controllable property, a deliberately conservative curvature-aware acceptance control scales an Adam proposal below the positive local quadratic root only when a separate frozen guard set predicts a reversal. A fixed 20% root margin and lower scale bound of $1/16$ are used across hard Burgers and Allen-Cahn. Five seeds and two matched controls, Adam and half-rate Adam, are run for 6000 FP64 updates. The guard set is distinct from training, audit, and solution evaluation.

Across 10 matched PDE-seed blocks, the control reduced median reversal prevalence to 3.44% from 11.61% under Adam and 8.59% under half-rate Adam. Median conditional risk fell to 5.62% from 18.83% and 14.29%, respectively. All four reversal comparisons remained significant after Holm correction ($p = 0.0156$). Final relative L_2 comparisons were

statistically indistinguishable from Adam ($p_{\text{Holm}} = 1.000$) and half-rate Adam ($p_{\text{Holm}} = 1.000$). The result establishes direct control of the monitored reliability axis while keeping solver accuracy as a separate objective.

In hard Burgers, median final relative L_2 errors were 0.336, 0.103, and 0.147 for Adam, half-rate Adam, and the control, while reversals remained measurable. In hard Allen-Cahn, the corresponding medians were 0.993, 0.996, and 0.997, so that family serves as a stress test. The combined result is constructive: executed-step reliability is measurable and directly reducible, while final accuracy remains governed by representation, weighting, sampling, conditioning, optimization trajectory, and residual-to-solution correspondence.

In the 45-run robustness sweep, weight decay did not consistently improve reliability or final error across three PDEs and three decay strengths; exact adaptive/decay step decomposition passed the supplied unit test. This broadens the executed-step mechanism without treating AdamW as a tuned baseline.

5 Meaning and Scope of the Gap

Taylor's theorem and Hessian-vector products are established tools. The contribution is component-wise reliability of the complete stateful optimizer map on unchanged objectives. Direction methods study how a step is proposed; the fixed-set audit tests whether the final displacement realizes the local claim. Proposition 1 isolates that event, Proposition 2 supplies an optional certificate, and the experiments quantify how often practical PINN steps cross the boundary.

The curvature-to-linear ratio in Eq. (18) gives a direct scale for the observed failure. When the quadratic term is small, first-order signs are reliable; when it approaches the linear term, reversals become common. Gradient alignment and conflict resolution remain useful properties of a proposed direction, but they do not by themselves certify the full adaptive step. In particular, the ConFIG results should be read as a post-optimizer finite-step audit, not as a test of ConFIG's training-batch projection theorem.

The gap is not mathematically exclusive to PINNs, but its interpretation is PINN-specific: each component represents a physical constraint, changing collocation points can contaminate before/after measurements, and solver accuracy is not identical to residual descent. The auxiliary intervention shows that executed-step reliability is actionable, while the statistically indistinguishable final-error comparisons show that it is complementary to, rather than a surrogate for, global solution accuracy. This separation is consistent with physical-admissibility gates, where passing a monitored envelope does not certify task success [7].

Practical use. At sparse checkpoints, evaluate fixed points before and after the ordinary step, compute $\boldsymbol{g}_{k,t}^{\top}\boldsymbol{s}_t$, and use one Hessian-vector product for the curvature explanation. The audit reports prediction, outcome, and explanation without modifying training. This separation mirrors runtime-assurance boundaries between learned proposals and consequential execution [28].

5.1 Limitations

Evidence is restricted to one tanh MLP, four low-dimensional forward PDEs, five seeds, and the tested update mechanisms; broader architectures, inverse or high-dimensional problems, L-BFGS schedules, adaptive weighting, calibration, and intervention efficiency remain future work.

6 Conclusion

This work makes component-wise reliability of the executed optimizer map observable. The fixed-set audit distinguishes local directional promise from realized component progress and shows that directional curvature explains frequent reversals across the tested grid. The intervention confirms that this reliability axis is controllable, while solver accuracy remains a complementary system-level objective. The practical requirement is to audit the displacement actually executed before treating a favorable local direction as realized progress.

A Supporting Derivations

For iid points, let $X = f_k^-(Z)$ and $Y = f_k^+(Z)$, with $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$. For n shared points,

$$\mathrm{Var}(\widehat{\Delta}_{\text{pair}}) = (\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})/n.$$

Independent before/after sets omit the covariance term.

A directional path-variation estimate is $\widehat{\rho}_{k,t}^{\text{dir}} = \max_j |\boldsymbol{s}_t^{\top}[\boldsymbol{H}_{k,t}(\alpha_j) - \boldsymbol{H}_{k,t}(0)]\boldsymbol{s}_t|/(\alpha_j \|\boldsymbol{s}_t\|_2^3)$, requiring one additional Hessian-vector product per probe.

For $\phi_{k,t}(\alpha) = \mathcal{L}_k(\boldsymbol{\theta}_t + \alpha \boldsymbol{s}_t; \mathcal{S}_k^{\text{aud}})$, integrating $\phi_{k,t}''(\alpha) = h_{k,t}(\alpha)$ twice gives Eq. (12). Subtracting Eq. (11), then

applying the segment Lipschitz condition, yields

$$A_{k,t} - P_{k,t}^{(2)} = \int_0^1 (1-\alpha)\boldsymbol{s}_t^\top [\boldsymbol{H}_{k,t}(\alpha) - \boldsymbol{H}_{k,t}(0)]\boldsymbol{s}_t \, d\alpha,$$

$$|A_{k,t} - P_{k,t}^{(2)}| \le \rho_{k,t}\|\boldsymbol{s}_t\|_2^3 \int_0^1 \alpha(1-\alpha)\, d\alpha = \frac{\rho_{k,t}}{6}\|\boldsymbol{s}_t\|_2^3.$$

The factor is exact because $\int_0^1 \alpha(1-\alpha)\, d\alpha = 1/2 - 1/3 = 1/6$.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686-707, 2019.

[2] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 26548-26560.

[3] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3055-A3081, 2021.

[4] S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," *J. Comput. Phys.*, vol. 449, Art. no. 110768, 2022.

[5] P. Rathore, W. Lei, Z. Frangella, L. Lu, and M. Udell, "Challenges in training PINNs: A loss landscape perspective," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 42159-42191.

[6] T. S lużalec, D. Wójcik, C. Uriarte, M. Loś, A. Paszyńska, and M. Paszyński, "Reliable physics-informed neural networks for Navier-Stokes simulations: Can we trust AI-generated numerical simulations?" *J. Comput. Sci.*, vol. 95, Art. no. 102817, 2026.

[7] B. Or, "Can predicted dynamics exist in the physical world?" arXiv:2606.00089, 2026.

[8] R. Bischof and M. A. Kraus, "Multi-objective loss balancing for physics-informed deep learning," *Comput. Methods Appl. Mech. Eng.*, vol. 439, Art. no. 117914, 2025.

[9] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 527-538.

[10] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 5824-5836.

[11] Y. Hwang and D.-Y. Lim, "Dual cone gradient descent for training physics-informed neural networks," arXiv:2409.18426, 2024.

[12] Q. Liu, M. Chu, and N. Thuerey, "ConFIG: Towards conflict-free training of physics-informed neural networks," in *Proc. Int. Conf. Learn. Representations*, 2025.

[13] S. Wang, A. Bhartari, B. Li, and P. Perdikaris, "Gradient alignment in physics-informed neural networks: A second-order optimization perspective," in *Adv. Neural Inf. Process. Syst.*, vol. 38, 2025.

[14] F. Liu, R. Saluja, S. Kwak, R. Wang, R. Deng, H. Kim, J. C. Paetzold, and M. R. Sabuncu, "MAdam: Metric-aware multi-objective Adam," arXiv:2606.03904, 2026.

[15] R. Z. Moayedi, M. Abbaszadeh, and M. Dehghan, "CuPINN: Optimizing PINNs through curvature minimization and residual landscape flattening," *Comput. Methods Appl. Mech. Eng.*, vol. 445, Art. no. 118180, 2025.

[16] E. Kiyani, K. Shukla, J. F. Urbán, J. Darbon, and G. E. Karniadakis, "Optimizing the optimizer for physics-informed neural networks and Kolmogorov-Arnold networks," *Comput. Methods Appl. Mech. Eng.*, vol. 446, Art. no. 118308, 2025.

[17] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[18] B. A. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Comput.*, vol. 6, no. 1, pp. 147-160, 1994.

[19] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, no. 153, pp. 1-43, 2018.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249-256.

[21] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 4, pp. 86-112, 1967.

[22] A. B. Owen, "Scrambling Sobol' and Niederreiter-Xing points," *J. Complexity*, vol. 14, no. 4, pp. 466-489, 1998.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[24] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223-311, 2018.

[25] B. Or, "Automatic stability and recovery for neural network training," arXiv:2601.17483, 2026.

[26] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80-83, 1945.

[27] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Stat.*, vol. 6, no. 2, pp. 65-70, 1979.

[28] B. Or, "Silent failures in physical AI: A literature review of runtime action authorization for autonomous systems," arXiv:2606.00090, 2026.

[29] J. C. Wong, A. Gupta, C. C. Ooi, P.-H. Chiu, J. Liu, and Y.-S. Ong, "Evolutionary optimization of physics-informed neural networks: Evo-PINN frontiers and opportunities," *IEEE Comput. Intell. Mag.*, vol. 21, no. 1, pp. 16-36, Feb. 2026, doi: 10.1109/MCI.2025.3607749.