SLED: SOBOL–LEVERAGE DESIGNS FOR THE COLLOCATION BATCH OF QUASI-NEWTON PHYSICS-INFORMED NEURAL NETWORK TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

A *two-phase optimization schedule* for physics-informed neural network training, a first-order *warmup* followed by a quasi-Newton optimizer, achieves more accurate results than single-phase approaches in a growing number of benchmarks. The quasi-Newton phase minimizes the objective on a collocation batch that is usually held fixed, and no pipeline selects that batch from the tangent features of the residual after the warmup. For the network linearized there, every gradient and Broyden-class method on a fixed batch returns the minimum-norm least-squares fit of that batch. The network leaves this linearization, but on all six benchmarks we study the linearized fit still orders the frozen batches by their loss off the batch, and a batch is judged by that loss. We build the batch once, after the warmup. Specifically, the Sobol–leverage design (SLED) combines a scrambled-Sobol set with points drawn from the ridge leverage of the residual's tangent features, in a share computed on the candidate pool. The ridge-leverage sampling bound, stated for these features, gives a sufficient batch size for a leverage batch, and the two benchmarks on which leverage points alone fail are the two on which the batch lies below it. On six benchmarks, SLED reaches a median error at or below that of three established frozen batches, the uniform default among them, and of three refresh protocols, up to $60\times$ and $375\times$ below the default, at 2.1–4.2% of the cost of the phase. It reaches the default's error at $16{,}384$ points with 2–$8\times$ fewer points on four of them. Each component alone trails it somewhere: leverage points fail on two benchmarks, and the Sobol set is 32–$82\times$ less accurate on two others.

1 INTRODUCTION

Physics-informed neural networks (PINNs) reformulate the solution of partial differential equations (PDEs) as an optimization of a residual-based objective evaluated on a neural surrogate (Raissi et al., 2019). For PINN training, a *two-phase optimization schedule* can achieve more accurate results than single-phase approaches in a growing number of benchmarks and is receiving increasing attention in the literature (Rathore et al., 2024; Urbán et al., 2025; Kiyani et al., 2025). Concretely, the schedule consists of (a) a first phase, the *warmup*, in which a faster-acting first-order optimizer trains the surrogate, and (b) a second phase, in which a more powerful quasi-Newton (QN) optimizer, started from the parameters at the end of the warmup, further minimizes the objective. In the second phase, the QN optimizer minimizes a weighted mean of the squared residuals over a batch of collocation points. The pipeline's *domain importance adaptation method* governs which points enter the batch and how they are weighted. We work in a highly competitive pipeline: on four of our six benchmarks, even a uniform batch held fixed through the second phase reaches errors below the best published ones on the same equations (Appendix C.5, for orientation only, since the pipelines are not matched).

We call a batch of collocation points drawn once and held fixed for the whole second phase a *frozen collocation set*, and we write *frozen batch* for short. Two reasons favor a frozen batch. First, the curvature model of a QN optimizer assumes a fixed objective, and a batch redrawn during the phase changes that objective; for this reason, Visser et al. (2025) hold their points fixed during each L-BFGS block. Second, a frozen batch makes the linearized second phase analyzable (Lemma 3.1).

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.

Existing two-phase training pipelines keep the same domain importance adaptation method for both phases. The second-phase collocation set is therefore whatever that method produces: the set inherited from the warmup, or one redrawn or reweighted during the phase. In the first case, the second phase runs on a frozen batch. Wu et al. (2023) train Sobol and Hammersley sets through an L-BFGS phase, and Andersen & Matsubara (2026) report overfitting to a frozen batch under single-phase L-BFGS training as a divergence of training and held-out losses. In the second case, four pipelines redraw the collocation set or reweight its points during the phase (Urbán et al., 2025; Kiyani et al., 2025; Toscano et al., 2026b; Jnini et al., 2026). However, no pipeline selects the frozen second-phase set from the tangent features of the residual after the warmup. A set inherited from a residual-based method, frozen RAD in Section 5.1, is drawn from the residual alone, and PINNACLE (Lau et al., 2024) re-selects its points from the tangent kernel during a first-order phase.

Taken together, these works leave three questions about the collocation set of the second phase open: (1) which solution the second phase returns on a frozen batch; (2) which points a frozen batch should contain; and (3) whether computation spent once after the warmup can replace redrawing the collocation set during the phase.

Against this background, we choose which collocation points the second phase keeps frozen. Our contributions can be summarized as follows:

- We introduce the Sobol–leverage design (SLED), a domain importance adaptation method that builds the set from a scrambled-Sobol set and ridge-leverage points of the residual's tangent features, with the share of the Sobol set computed on the run's own candidate pool. In our benchmarks, its median error is at or below that of every established method, up to $60\times$ and $375\times$ below a frozen uniform default, at 2.1–4.2 % of the cost of the second phase, and it reaches that default's error with 2–$8\times$ fewer points on four benchmarks.
- We conduct a matched comparison of ten methods from shared per-seed warmup checkpoints: a frozen uniform default, frozen RAD, frozen PINNACLE-K, three published refresh protocols, and the ablations of SLED. That is, each component alone, and the score replaced by the kernel diagonal. A batch chosen once after the warmup matches or outperforms every refresh protocol tested.
- We give the analysis behind the design: for the network linearized at the end of the warmup, every method satisfying the linear-span assumption, L-BFGS and self-scaled Broyden included, returns the minimum-norm least-squares fit of its frozen batch (Lemma 3.1, by a classical argument). The network leaves this linearization, but the fit still orders the frozen batches by their off-batch loss on all six benchmarks (Appendix D), so a batch is judged by that loss. The ridge-leverage embedding bound of Bach (2017), stated for the tangent features and carried to the error of the interpolant, gives a sufficient batch size for a leverage batch; it lies above the batch on the two benchmarks where leverage points alone fail, and we use it as a consistency check, not as a prediction.

2 RELATED WORK

Domain importance adaptation methods for the first phase fall into two groups, those that choose the points and those that weight them. Within the first group, residual-based adaptive refinement adds points where the residual is largest (Lu et al., 2021), residual-based adaptive distribution (RAD) samples a residual-proportional density (Wu et al., 2023), retain-resample-release sampling retains high-residual points and resamples the rest (Daw et al., 2023), and Langevin adaptive sampling draws points by Langevin dynamics on the residual (Jeong et al., 2025). For the weighting, variational residual-based attention (vRBA) derives residual weights as the optimal adversary of an entropy-regularized L^∞ objective (Toscano et al., 2026b), and causal weighting imposes a temporal order (Wang et al., 2024). However, none of these methods addresses which set a second phase keeps frozen.

For the second phase, Urbán et al. (2025) introduce self-scaled Broyden (SSBROYDEN) methods into PINN training, and Kiyani et al. (2025) study the family systematically. We refer to a pipeline that refreshes the collocation set during the second phase, that is, redraws it or reweights its points, as a *refresh protocol*. Three refresh protocols have been proposed: uniform draws every 500 iterations (Urbán et al., 2025), a RAD density every 500 iterations (Kiyani et al., 2025) and a vRBA-weighted batch every 100 iterations (Toscano et al., 2026b). Visser et al. (2025) move their points during Adam and hold them fixed for each L-BFGS block. However, apart from the held-out loss of Andersen &

Matsubara (2026), none of these pipelines reports the off-batch loss, the training objective evaluated at points outside the batch (defined in Section 3), and none uses it to choose the set.

In the approximation literature, sampling by the inverse Christoffel function stabilizes polynomial least squares (Cohen et al., 2013; Cohen & Migliorati, 2017; Adcock, 2025). Similarly, ridge leverage scores play the same role for kernel regression (El Alaoui & Mahoney, 2015; Cohen et al., 2017; Rudi et al., 2018). Specifically, sampling from an overestimate of the ridge-leverage density embeds the ridge-shifted Gram with a number of points proportional to the effective dimension, up to a logarithm, whereas uniform sampling needs the supremum of the leverage function (Bach, 2017; Avron et al., 2017; Musco & Musco, 2017). Theorem 4.2 is that embedding bound stated for the tangent features of the residual at the end of the warmup.

Within PINN training, Adcock et al. (2023) sample collocation points from the Christoffel function of the network's last-layer span, and Lau et al. (2024) select them by sampling or by K-means++ seeding (PINNACLE-K) from the residual projected onto Nyström eigenfunctions of the tangent kernel on a pool; both re-select their points during a first-order phase and score with the plain feature span or the residual, respectively, not with the ridge leverage of the operator-applied tangent features. In batch active learning for regression, the kernel diagonal, ridge leverage and the power function of gradient features are standard selection scores (Holzmüller et al., 2023). The Sobol component of SLED has its own antecedent, low-discrepancy collocation for PINNs with error bounds in the discrepancy (Mishra & Rusch, 2021; Longo et al., 2021). The RAD density already mixes a residual score with a uniform component (Wu et al., 2023), the defensive mixture of importance sampling (Hesterberg, 1995). SLED differs in the score, in the Sobol component and in the Sobol share computed on the pool. Concurrent with this work, Linghu & Wang (2026) introduce residual-Christoffel sampling for random-feature collocation of linear PDEs. The correspondence is close: their rank-truncated Christoffel density plays the role of our leverage density, their retained dimension that of our effective dimension, and their Gram-concentration theorem is Theorem 4.2 in a different feature space (Section 4.1). We differ in the feature space, the operator-applied tangent space of a trained nonlinear network rather than a random feature space drawn once, and in characterizing what the second phase returns.

3 SETTING: THE SECOND PHASE ON A FROZEN BATCH

Two-phase training and the frozen batch. Let $D \subset \mathbb{R}^d$ be a bounded space–time domain, ρ the uniform probability measure on it, and $L^2(\rho) = L^2(D, \rho)$ with norm $\|\cdot\|_\rho$. The first-order warmup of Section 1 produces parameters $\theta_0 \in \mathbb{R}^P$ of a network u_θ with PDE residual $r(\cdot\,; \theta) : D \to \mathbb{R}$. The second phase then minimizes

$$\widehat{L}(\theta) \;=\; \frac{1}{n} \sum_{i=1}^{n} \omega_i \, r(x_i; \theta)^2 \tag{1}$$

over a frozen batch $C = \{x_1, \dots, x_n\} \subset D$ with fixed weights $\omega_i > 0$. We write *on-batch loss* for $\widehat{L}$ at the points of the batch and *off-batch loss* for the same objective evaluated at points outside the batch. Remark A.3 treats the initial-condition term and the boundaries, Table 5 reports row and parameter counts, and Section 5 gives the benchmarks and the budget of the second phase.

Tangent features and the feature operator. We write $r_0 := r(\cdot\,; \theta_0)$ and $\varphi(x) := \nabla_\theta r(x; \theta_0) \in \mathbb{R}^P$. The *feature operator* $\Phi : \mathbb{R}^P \to L^2(\rho)$, $(\Phi\delta)(x) = \varphi(x)^\top \delta$, has adjoint $\Phi^* g = \mathbb{E}_\rho[\varphi\, g]$ and second-moment matrix $\mathbf{G} = \Phi^*\Phi = \mathbb{E}_{x\sim\rho}[\varphi(x)\varphi(x)^\top]$. We write $\sigma_1^2 \geq \cdots \geq \sigma_R^2 > 0$ for the eigenvalues of $\mathbf{G}$ and $R := \operatorname{rk}\Phi$ (its singular value decomposition: eq. (4), Appendix A). The empirical neural tangent kernel (NTK) of the residual (Jacot et al., 2018; Wang et al., 2022) is $K(x, y) = \varphi(x)^\top \varphi(y)$, with diagonal $\kappa(x) = \|\varphi(x)\|_2^2$.

For the second-phase trajectory $(\theta_t)_{t\geq 0}$, writing $\delta_t = \theta_t - \theta_0$, we write the residual along the phase as

$$r(x; \theta_0 + \delta_t) \;=\; r_0(x) + \varphi(x)^\top \delta_t + \eta_t(x), \qquad \varepsilon_{\mathrm{lin}} \;:=\; \sup_{t\geq 0} \|\eta_t\|_\rho \,, \tag{2}$$

and Lemma 3.1 is exact for the linear model $r_0 + \Phi\delta$, that is, for $\eta_t \equiv 0$. We say that an iterative method for $\widehat{L}$ with iterates $\delta_0 = 0, \delta_1, \dots$ and gradients $g_t = \nabla\widehat{L}(\delta_t)$ satisfies the *linear-span assumption* (Nesterov, 2018) if $\delta_{t+1} \in \operatorname{span}(\{\delta_0, \dots, \delta_t\} \cup \{g_0, \dots, g_t\})$ for every t. Lemma 3.1 says what the second phase returns.

Lemma 3.1 (For the linearized model, the batch fixes the limit)**.** *Let $\widehat{L}$ be the objective* (1) *for the linear model $r_0 + \Phi\delta$ (eq.* (2) *with $\eta_t \equiv 0$), let $V_C := \operatorname{span}\{\varphi(x_i) : x_i \in C\}$, let $n_{\mathrm{d}} \le n$ be the number of distinct points of C, let Φ_C have rows $\varphi(x_i)^\top$ over those points, and let W be the $n_{\mathrm{d}} \times n_{\mathrm{d}}$ diagonal matrix whose entry for a distinct point is the sum of the weights ω_i of its copies in C.* (i) Iterates in the batch span. *Every method satisfying the linear-span assumption has $\delta_t \in V_C$ for all t. The class contains gradient descent, momentum, Nesterov, nonlinear conjugate gradients, and every Broyden-class method with $H_0 = \gamma I$ and any line search (Nocedal & Wright, 2006, Ch. 6), limited-memory BFGS and the self-scaled Broyden method of Section 5 included. Adam-type preconditioning is not in the class.* (ii) Independent rows. *If the n_{d} rows of Φ_C are linearly independent, the set of interpolants $\mathcal{A} = \{\delta : \Phi_C\delta = -r_0|_C\}$ intersects V_C in exactly one point, the minimum-ℓ_2-norm interpolant $\delta^{\mathrm{mn}} = -\Phi_C^+ r_0|_C = \arg\min\{\|\delta\|_2 : \delta \in \mathcal{A}\}$, with $(\cdot)^+$ the pseudoinverse. Consequently, if such a method attains $\widehat{L}(\delta_\infty) = 0$ in exact arithmetic, at an iterate or as the limit of its iterates, then $\delta_\infty = \delta^{\mathrm{mn}}$, independently of the optimizer, its hyperparameters, and the weights ω. The updated residual field is then the error of the kernel interpolant of r_0 on C (eq.* (7)*, Appendix B).* (iii) Inconsistent rows. *If $\mathcal{A}$ is empty, then a method satisfying the linear-span assumption that attains the minimum of $\widehat{L}$, at an iterate or as the limit of its iterates, returns the minimum-norm weighted least-squares solution $\delta^{\mathrm{ls}} = -(W^{1/2}\Phi_C)^+ W^{1/2} r_0|_C$ in place of δ^{mn}. The weights enter the fit, and the optimizer still does not.*

The proof, with the antecedents of the lemma (Wilson et al., 2017; Gunasekar et al., 2018; Nazareth, 1979; Jacot et al., 2018), is given in Appendix B. The empty case of part (iii) arises when the rows of every residual family outnumber the parameters (Remark A.3). Numerically, it is the case on every frozen batch of our experiments, in one space dimension too (Appendix D). By part (i), the linearized second phase moves θ_0 only within the span of the batch features, so it leaves the residual after the warmup almost unchanged on every region whose tangent features are nearly orthogonal to that span (Remark A.6).

4 Choosing the batch: a sampling bound and the Sobol–leverage design

Section 4.1 gives the batch sizes above which a leverage batch and a uniform batch embed the regularized Gram, and Section 4.2 specifies the Sobol–leverage design (SLED).

4.1 The sampling bound: ridge leverage of the tangent features

A *batch rule* is a density q on D: the batch is drawn from q and weighted by $\omega = 1/q$. The question of this subsection is which batch sizes n suffice for such a batch, and for a uniform batch, to embed $\mathbf{G} + \lambda I$ in the sense of (3). Definition 4.1 and Theorem 4.2 answer it.

Definition 4.1 (Ridge leverage and effective dimension (El Alaoui & Mahoney, 2015; Musco & Musco, 2017))**.** For $\lambda > 0$,

$$\ell_\lambda(x) \;:=\; \varphi(x)^\top(\mathbf{G} + \lambda I)^{-1}\varphi(x), \qquad d_\lambda \;:=\; \mathbb{E}_\rho[\ell_\lambda] \;=\; \operatorname{tr}\big(\mathbf{G}(\mathbf{G} + \lambda I)^{-1}\big) \;=\; \sum_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda}.$$

The *leverage density* is $q_\lambda^{\mathrm{lev}} := \ell_\lambda / d_\lambda$. For $\beta \in (0, 1]$, a density q is a *β-leverage-overestimate* if $q(x) \;\ge\; \beta\, q_\lambda^{\mathrm{lev}}(x)$ for ρ-a.e. x.

Theorem 4.2 (Spectral embedding of the weighted Gram)**.** *Let q be a β-leverage-overestimate, let $x_1, \dots, x_n$ be i.i.d. from $q\,\mathrm{d}\rho$, let $\omega := 1/q$, let $\widehat{\mathbf{G}} = \frac{1}{n}\sum_i \omega(x_i)\varphi(x_i)\varphi(x_i)^\top$, and let $\varepsilon, \zeta \in (0, 1)$. If $\lambda \le \lambda_{\max}(\mathbf{G})$ and $n \ge 4\varepsilon^{-2}(d_\lambda/\beta)\log(16 d_\lambda/\zeta)$, then, with probability at least $1 - \zeta$ over the draw,*

$$(1 - \varepsilon)\,(\mathbf{G} + \lambda I) \;\preceq\; \widehat{\mathbf{G}} + \lambda I \;\preceq\; (1 + \varepsilon)\,(\mathbf{G} + \lambda I). \tag{3}$$

Theorem 4.2 is the ridge-leverage sampling bound of Bach (2017), Avron et al. (2017), and Musco & Musco (2017), stated for the tangent features of the residual at the end of the warmup; the polynomial, $\lambda = 0$ case is Cohen & Migliorati (2017), and the proof is given in Appendix B. The uniform density $q \equiv 1$ is a β-leverage-overestimate with $\beta = d_\lambda / \sup_x \ell_\lambda$, so the theorem covers a uniform batch with the prefactor $d_\lambda/\beta = \sup_x \ell_\lambda$.

Algorithm 1 The Sobol–leverage design (SLED)

1: **Input and pool:** θ_0, n; ridge $\lambda = \lambda_{\mathrm{rel}}\widehat{\lambda}_{\max}$, $\lambda_{\mathrm{rel}} = 10^{-4}$; $S = \{z_j\}_{j\le M} \overset{\text{iid}}{\sim} \rho$, $M = \max\{4n,\ 65{,}536\}$; Nyström $\widehat{\ell}_\lambda(z_j)$ ($L = 1{,}024$ landmarks, $k = 256$ eigenpairs); Hutchinson $\widehat{\kappa}(z_j)$ ($s = 8$ probes)
2: **Sobol share:** $\mu_{\mathrm{sob}} \leftarrow \widehat{\rho} := \max\{0, \mathrm{Spearman}(\widehat{\ell}_\lambda, \widehat{\kappa})\}$ on the pool
3: **Sobol component:** a scrambled Sobol set of $\mu_{\mathrm{sob}} n$ points, weights $\omega \equiv 1$
4: **Leverage points:** $(1-\mu_{\mathrm{sob}})n$ pool indices $\sim q_j = \widehat{\ell}_\lambda(z_j)/\sum_k \widehat{\ell}_\lambda(z_k)$, with replacement and with no uniform mixture component; weights $\omega_i = 1/(Mq_i)$
5: **Output:** the frozen batch (Sobol component $\cup$ leverage points) with its weights

We call the lower bound on n in Theorem 4.2 the *sample-complexity bound* of a batch. With $\beta = 1$ and the constants $\varepsilon = \frac{1}{2}$ and $\zeta = 0.1$, the sample-complexity bound of a batch drawn from the leverage density (a *leverage batch*) is

$$n_{\mathrm{Lev}} \ := \ 4\varepsilon^{-2}\, d_\lambda \log(16 d_\lambda/\zeta),$$

and that of a uniform batch is the same expression with $\sup_x \ell_\lambda$ in place of the prefactor d_λ. The two bounds differ by the factor $\sup \ell_\lambda / d_\lambda \ge 1$, the maximum of the leverage function over its mean, which we call the *coherence* of the tangent features, the ridge analogue of the coherence of Candès & Recht (2009). All three are estimated on a pool (Remarks A.2 and A.8). The uniform bound is the classical coherence condition of unweighted least squares, sharp for polynomial spaces (Cohen et al., 2013).

The following corollary carries the embedding to the error of the interpolant. For $f = \varphi^\top \delta$ in the space $H_K = \{\varphi(\cdot)^\top \delta : \delta \in \mathbb{R}^P\}$ with norm $\|f\|_{H_K} = \min\{\|\delta\|_2 : f = \varphi^\top \delta\}$, let $s_C f := \varphi^\top \Pi_C \delta$ be its interpolant on C, where Π_C is the orthogonal projector onto V_C; with $k_C(x) = (K(x, x_i))_i$ and $\mathbf{K}_C = (K(x_i, x_j))_{ij}$ this is the kernel interpolant $k_C(\cdot)^\top \mathbf{K}_C^+ f|_C$, with $(\cdot)^+$ the pseudoinverse (Appendix B).

Corollary 4.3 (Interpolation error on the embedding event)**.** *In the setting of Theorem 4.2, on the event that the lower bound in* (3) *holds, every $f \in H_K$ satisfies*

$$\|f - s_C f\|_\rho \ \le \ \sqrt{\varepsilon\lambda/(1-\varepsilon)}\ \|f\|_{H_K}.$$

The bound does not depend on the weights, and it holds for every batch that contains the batch of the theorem.

The proof is given in Appendix B. *Scope of the theory.* Lemma 3.1 is interpreted for the network as a statement about its linearization at the end of the warmup, and $\varepsilon_{\mathrm{lin}}$ is not bounded. Appendix D measures it at the end of the phase on two benchmarks. Applied to the part of r_0 in the range of Φ, Corollary 4.3 bounds the residual the linearized phase returns (Lemma 3.1(ii)) for every leverage batch above n_{Lev}, by $\sqrt{\lambda}\ \|\cdot\|_{H_K}$ at $\varepsilon = \frac{1}{2}$. The part of r_0 outside that range is not bounded, and the interpolating phase is the limit $\lambda \to 0$ of the family the theorem describes (Remark A.5). Both bounds are sufficient conditions, and nothing is guaranteed below them. Section 5.3 sets the point of diminishing returns of a leverage batch (Section 5.1) against n_{Lev}.

4.2 The Sobol–leverage design: leverage points and a scrambled-Sobol component

SLED builds the frozen batch after the warmup in three steps: (i) it scores a pool of M points drawn i.i.d. from ρ by the Nyström estimate $\widehat{\ell}_\lambda$ of the ridge leverage of Definition 4.1; (ii) it draws $(1-\mu_{\mathrm{sob}})n$ *leverage points* from the pool with replacement, with probability $q_i \propto \widehat{\ell}_\lambda(z_i)$ and weights $\omega_i = 1/(Mq_i)$; (iii) it adds a scrambled Sobol set of $\mu_{\mathrm{sob}} n$ points at weight one, the *Sobol component* C_{cov}. The *Sobol share* μ_{sob} is the rank correlation $\widehat{\rho}$ of $\widehat{\ell}_\lambda$ with the estimate $\widehat{\kappa}$ of the kernel diagonal on the pool (line 2), and it is not tuned.

The score of SLED uses only the tangent features. The interpolation error on a scrambled Sobol set is bounded through its *fill distance* (Lemma A.9), a quantity of the point set.

By Proposition A.4(i), the density $\kappa/\operatorname{tr}\mathbf{G}$ of the kernel diagonal is a β_κ-leverage-overestimate, and β_κ decays as the spectrum of $\mathbf{G}$ steepens. SLED therefore draws its leverage points from $\widehat{\ell_\lambda}$ and not from $\widehat{\kappa}$. By Proposition A.4(iii), $\widehat{\rho}$ is the rank agreement between the leverage ranking of the pool at the ridge λ and its ranking in the limit $\lambda \to \infty$, the ranking by the kernel diagonal. If the two rankings agree ($\widehat{\rho}$ near one), no direction is localized at the scale resolved at the end of the warmup and a space-filling design suffices, so the batch goes to the Sobol component; if they disagree ($\widehat{\rho}$ near zero), localized directions exist and need leverage points. The Sobol share interpolates between the two cases, and SLED uses $\widehat{\rho}$ as the share directly, with no further constant. We note that $\widehat{\rho}$ is a rank statistic. That these rankings stay stable below λ, along the phase, is Assumption 1 (Appendix A.7). We refer to the batches at $\mu_{\mathrm{sob}} = 1$, a Sobol set, and at $\mu_{\mathrm{sob}} = 0$, leverage points alone, as the *pure Sobol batch* and the *pure leverage batch* of SLED, and to the one with the lower error on a benchmark as the *better component*. The batch rule LEV of Section 5 draws from a mixture of the leverage density (weight 0.8) and the uniform density (weight 0.2), a *uniform mixture component*, and is our proxy for the pure leverage batch.

The following proposition says what the batch of SLED keeps of its Sobol component.

Proposition 4.4 (Power function of a union of batches)**.** *With H_K, s_C, k_C and $\mathbf{K}_C$ as in Section 4.1, let $P_C(x)^2 := \kappa(x) - k_C(x)^\top \mathbf{K}_C^+ k_C(x)$ be the power function of the batch C.* (i) *For every $f \in H_K$ and $x \in D$, $|f(x) - (s_C f)(x)| \le P_C(x)\ \|f\|_{H_K}$.* (ii) *$P_C$ is pointwise non-increasing under inclusion of batches. In particular, for the Sobol component C_{cov} and the leverage points C_{lev} of Algorithm 1,*

$$P_{C_{\mathrm{cov}} \cup C_{\mathrm{lev}}}(x) \ \le\ \min\big\{P_{C_{\mathrm{cov}}}(x),\ P_{C_{\mathrm{lev}}}(x)\big\} \qquad \textit{for every } x \in D.$$

Part (i) is the classical power-function bound (Wendland, 2005, Thm. 11.4); both parts are proved in Appendix B. Part (ii) needs no i.i.d. assumption, so the fill-distance bound of the Sobol component (Lemma A.9) holds for the batch of SLED as a whole.

5 EXPERIMENTS

We conduct three tests, one per contribution and in their order (Section 1). Section 5.2 tests that the off-batch loss ranks the frozen batches and that the linearization of Lemma 3.1 predicts that ranking. Section 5.3 sets the sample-complexity bound of Theorem 4.2 against the batch size at which the final error stops improving. Section 5.4 compares SLED with its pure Sobol and pure leverage batches and with the established domain importance adaptation methods.

5.1 EXPERIMENTAL SETUP

The training pipeline consists of four components, the neural surrogate, the loss-balancing method, the optimization schedule and the domain importance adaptation method. Only the domain importance adaptation varies across our experiments. The six benchmarks are Allen–Cahn (AC) in the formulation of Kiyani et al. (2025), Burgers (Raissi et al., 2019), and Korteweg–de Vries (KdV), Kuramoto–Sivashinsky (KS), Ginzburg–Landau (GL) and Rayleigh–Taylor (RT) in the configurations of Wang et al. (2026). Four of them, AC, Burgers, KdV and KS, have one space dimension. GL and RT are systems in two space dimensions with two and four residual rows per point, respectively. The reported accuracy is the final relative L^2 error against a reference solution (Details in Appendix C.1).

The other three components are held fixed: a multilayer perceptron of five layers of width 64 with random Fourier features, a warmup of SOAP (Vyas et al., 2025; Wang et al., 2025) on $8{,}192$ uniform points redrawn every iteration, for $2{,}000$ iterations on the four benchmarks with one space dimension and $20{,}000$ on GL and RT, and GradNorm loss balancing (Wang et al., 2023) frozen after the warmup. For each seed, one warmup is run, and we refer to its parameters at the end of the warmup as the *warmup checkpoint* of that seed. Every second phase of a seed starts from its warmup checkpoint and runs for $40{,}000$ iterations of SSBroyden2 (Urbán et al., 2025) in float64 (PDEs, reference solutions and hyperparameters: Appendices C.1–C.3).

We compare ten methods, frozen or refreshed. All run in the same pipeline from the same per-seed warmup checkpoints with $n = 16{,}384$. We call a method *established* if it has been recently used for PINN training, the uniform default included, and six of the ten methods are established baselines. Three of them are frozen: the uniform default UNIF, RAD (Wu et al., 2023) with $k = 1$, and

Table 1: Final relative L^2 error at the end of the second phase for the six established methods, SLED and its ablations at $n = 16{,}384$ from the shared warmup checkpoints. Medians of ten seeds per method. Header rows: the error after the warmup and two pool quantities. Cells: median over seeds, spread $\max / \min$ in parentheses. Bold is the lowest median of a column and every cell within $1.3\times$ of it. * highlights below UNIF on every seed. $^{\dagger k}$ highlights k seeds above an error of 10^{-2}.

	Allen–Cahn	Ginzburg–Landau	Burgers	KdV	Kuramoto–Siv.	Rayleigh–T.
Computed after the warmup						
error after the warmup (median over seeds)	0.067	0.25	0.014	0.35	0.13	0.36
coherence $\sup \widehat{\ell_\lambda}/\widehat{d_\lambda}$ on the pool	12–20	93–219	126–428	11–39	61–134	23–32
$\widehat{\rho}$ (Sobol share of SLED) on the pool	0.00	0.40	0.58	0.88	0.93	0.98
Established, refresh protocols with period T_r (Urbán et al., 2025; Kiyani et al., 2025; Toscano et al., 2026b)						
RANDOM-R, T_r=500	4.1e-7 (21×)	3.0e-4* (1.5×)	2.5e-8 (10×)	1.1e-6 (1.4×)	1.6e-6 (3.3×)	1.3e-3 (2.4×)
RAD k=1, T_r=500	**1.2e-7** (1.4×)	2.7e-4* (1.4×)	9.6e-10* (3.8×)	1.2e-6 (1.4×)	2.9e-6 (3.7×)	2.5e-3 (2.3×)
vRBA weights, T_r=100	1.2e-6 (6.7×)	5.4e-3 (1.4×)	8.1e-8 (2e3×)	1.4e-3 (1.9×)	1.2e-1$^{\dagger 6}$ (1.0×)	1.4e-1$^{\dagger 6}$ (5.3×)
Established, frozen (Wu et al., 2023; Lau et al., 2024)						
uniform (UNIF, default)	9.3e-7 (141×)	3.2e-3 (5.9×)	1.2e-7 (67×)	**7.2e-7** (2.0×)	9.8e-7 (4.4×)	**5.4e-4** (7.0×)
RAD (k=1)	7.8e-7 (42×)	3.0e-4* (9.4×)	1.5e-9* (13×)	8.9e-7 (1.4×)	7.8e-6$^{\dagger 3}$ (8e4×)	7.7e-4$^{\dagger 4}$ (549×)
PINNACLE-K	2.2e-7 (47×)	3.7e-4* (2.5×)	**4.0e-10*** (4.3×)	2.5e-3$^{\dagger 1}$ (11×)	1.2e-1$^{\dagger 10}$ (1.6×)	4.7e-1$^{\dagger 10}$ (1.0×)
SLED *and its ablations (frozen)*						
SLED, $\mu_{\mathrm{sob}} = \widehat{\rho}$	**1.1e-7** (2.1×)	**5.1e-5*** (2.4×)	**3.2e-10*** (4.0×)	**5.8e-7** (2.2×)	3.7e-7* (2.2×)	**5.0e-4**$^{\dagger 1}$ (425×)
SOBOL(μ_{sob}=1)	4.3e-7 (8.5×)	1.5e-3 (2.1×)	2.3e-8 (16×)	**6.7e-7** (1.5×)	**2.2e-7*** (2.5×)	**4.9e-4** (1.8×)
LEV (0.2 uniform mixture)	1.5e-7 (11×)	**5.4e-5*** (2.0×)	**3.9e-10*** (9.0×)	**7.0e-7** (1.4×)	2.1e-2$^{\dagger 5}$ (2e4×)	2.8e-1$^{\dagger 9}$ (211×)
score $\widehat{\kappa}$ in place of $\widehat{\ell_\lambda}$ (KD-W)	7.5e-6 (14×)	1.1e-3 (16×)	8.6e-10* (15×)	1.8e-6 (9e3×)	1.1e-1$^{\dagger 10}$ (1.3×)	4.1e-1$^{\dagger 4}$ (1.2×)

PINNACLE-K (Lau et al., 2024), selected once on the warmup checkpoint with the authors' code and not re-selected. Three are the refresh protocols of Section 2: uniform and RAD redrawn every 500 iterations (Urbán et al., 2025; Kiyani et al., 2025), and vRBA redrawn every 100 iterations (Toscano et al., 2026b). The remaining four methods are SLED and its ablations: the Sobol component alone, a scrambled Sobol set, leverage points alone, LEV, which draws Nyström ridge-leverage points with a uniform mixture component of 0.2, and the score replaced by the kernel-diagonal estimate $\widehat{\kappa}$, KD-W. The densities of all methods are given in Table 7.

Table 1 reports the median of ten seeds per method with the spread $\max / \min$ in parentheses. A seed whose final error lies above 10^{-2} counts as failed. A ratio between two methods is paired on the shared warmup checkpoints: the median over seeds of the per-seed ratio, with a 95 % percentile bootstrap interval. Figure 1 uses six of the ten seeds (Appendix C.4). We refer to the smallest batch size run after which every later doubling improves the median error by less than a factor of two, at least one doubling following, as the *point of diminishing returns* of a method.

5.2 THE OFF-BATCH LOSS RANKS THE FROZEN BATCHES, AND THE LINEARIZATION PREDICTS THE RANKING

Table 2: Residual loss at the end of the second phase for the frozen methods of Table 1 at $n = 16{,}384$. First row: on the batch itself, median over the six methods' seed medians with their spread $\max / \min$. Middle rows: on 65,536 held-out points drawn i.i.d. uniformly on the domain, the unweighted mean of the squared residual, median over seeds. Last row: Pearson correlation of $\log_{10}$ off-batch loss with $\log_{10}$ final error over all runs of the six methods.

	Allen–Cahn	Ginzburg–Landau	Burgers	KdV	Kuramoto–Siv.	Rayleigh–T.
on-batch loss, seven methods (spread)	8.9e-17 (7×)	9.9e-13 (11×)	3.4e-17 (30×)	1.0e-12 (223×)	1.8e-12 (3×)	5.5e-10 (2e5×)
off-batch loss on 65,536 held-out points						
UNIF	2.3e-12	3.2e-5	1.3e-12	1.0e-11	5.7e-10	5.9e-7
RAD k=1	8.8e-13	5.1e-7	2.6e-16	1.1e-11	5.3e-10	3.7e-7
PINNACLE-K	2.4e-13	5.1e-9	4.2e-17	1.8e-3	8.9e-2	2.9e-4
SOBOL	2.9e-13	2.3e-5	1.6e-13	2.0e-12	6.0e-11	2.5e-7
LEV	2.5e-14	1.7e-8	2.8e-17	2.4e-12	9.2e-6	3.6e-6
KD-W	6.6e-10	7.1e-6	7.9e-17	4.5e-9	2.7e-2	3.0e-5
SLED	6.6e-15	9.9e-9	3.0e-17	1.9e-12	5.6e-12	2.8e-7
r, $\log_{10}$ off-batch loss vs. $\log_{10}$ final error	0.91	0.81	0.96	0.99	0.97	0.77

In Table 2, the on-batch loss is common to the methods of a benchmark up to the printed spread, while the off-batch loss is set by the batch: within a benchmark it spans several decades across the methods,

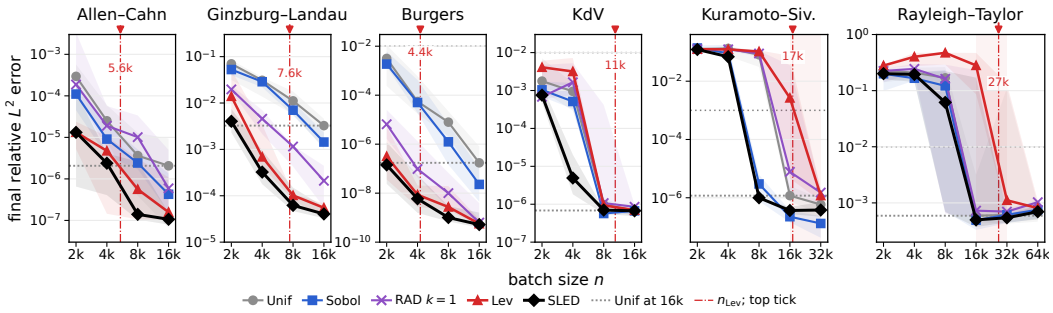

Figure 1: Final relative L^2 error (median over seeds; shaded, minimum to maximum) against the batch size n for five frozen methods on the six benchmarks. red dash-dotted, the sample-complexity bound n_{Lev} with them (Section 4.1). Dense grey dotted: the median error of UNIF at 16,384. The red strip marks the doubling in which the median of LEV reaches it. Detailed results per batch size are provided in Appendix C.4.

and it predicts the final error on every benchmark. Every batch system is numerically rank-deficient and inconsistent, so Lemma 3.1(iii) applies (Appendix D). The linearized fit at θ_0 overestimates the realized off-batch loss by two to five orders of magnitude on five benchmarks and by $6\times$ on RT, yet it orders the frozen batches as the realized loss does, with a median Spearman correlation across the methods of 0.70–0.94 on the six benchmarks; it orders the final errors on GL only (six frozen methods, three seeds; Appendix D). Hence a domain importance adaptation method for the second phase can be judged by its off-batch loss, which the tangent features at θ_0 already rank.

5.3 BATCH-SIZE SCALING

Figure 1 varies the batch size of five frozen methods. Theorem 4.2 guarantees the embedding for a leverage batch above its sample-complexity bound n_{Lev}. With the theorem's constants, the uniform bound lies above every batch size run, so only n_{Lev} is set against a point of diminishing returns (Remark A.8). On AC, GL and Burgers, LEV has a lower error than UNIF at every batch size run. Wherever the batch sizes run locate the point of diminishing returns of LEV, on GL, KdV and RT, it lies within one doubling of n_{Lev} (Table 8). On KdV and RT those of the other methods lie within one doubling of n_{Lev} as well. Only GL separates the methods. On AC and Burgers, LEV still improves by more than a factor of two at 16,384, so its point of diminishing returns lies more than one doubling above n_{Lev}. On KS and RT, n_{Lev} lies above 16,384, and LEV fails there on both. It reaches the cutoff one doubling later, its median error falling by three decades on KS and by two on RT over that doubling (Appendix C.4).

Both bounds are sufficient conditions, so their ratio does not fix which method reaches the cutoff first: on KS and RT it puts n_{Lev} well below the uniform bound, yet UNIF reaches the cutoff at 16,384 and LEV one doubling later. Above 16,384, SLED lies below LEV on every seed, and on RT the error of every method rises slightly with the batch size, with the pool grown alongside (Appendix C.4).

Table 3: SLED compared to its components below $n = 16{,}384$. First row: the smallest n at which the median of SLED lies at or below that of UNIF at 16,384, with the factor of points. Following rows: the median of the better component divided by that of SLED at the same n. A factor above one means SLED is lower. fail: the error of all three methods lie above 10^{-2}. RT is omitted because every method fails for both batch sizes.

better component	AC LEV	GL LEV	Burgers LEV	KdV SOBOL	KS SOBOL
n for SLED to reach UNIF at 16,384	8,192 (2×)	4,096 (4×)	2,048 (8×)	16,384 (1×)	8,192 (2×)
better / SLED, $n = 4{,}096$	2.0×	2.1×	1.5×	$10^2\times$	fail
better / SLED, $n = 8{,}192$	4.0×	1.6×	2.7×	0.8×	3.0×

At 16,384, SLED is close to its better component on every benchmark acting more as a switch between the two. Below 16,384 this separates. Table 3 compares the medians of Table 8 at the fixed batch

sizes $4{,}096$ and $8{,}192$ on the five benchmarks on which not every method fails at both sizes. SLED lies below that of the better component in eight of the nine cases that do not fail, by 1.5–4×, and by about $10^2\times$ on KdV at $4{,}096$. One exception is KdV at $8{,}192$, where SOBOL lies $1.2\times$ lower. In the median, SLED reaches the error of UNIF at $16{,}384$ with 2–8× fewer points on AC, GL, Burgers and KS, and needs $16{,}384$ points on KdV.

5.4 SLED AT A FIXED BATCH SIZE

Table 1 compares SLED with the six established methods and with its own ablations at $n = 16{,}384$. Appendix C.8 (Figures 2–5) draws the batches of one seed on the four one-dimensional benchmarks. The failed cells of Table 1 are the batches that concentrate in one band and leave the rest of the domain uncovered. SLED lies below the frozen default UNIF on AC, GL, Burgers and KS, at paired ratios of 0.21, 0.02, 0.00 and 0.37, respectively, with every interval excluding one (Table 11); on KdV and RT the intervals contain one. Against frozen RAD it lies below on every benchmark. Against the three refresh protocols, SLED is at or below each of them in the median of every column. This answers the third question of Section 1: a batch chosen once after the warmup can match or outperform every refresh protocol tested. Against the best established method per benchmark, SLED is about $5\times$ better on GL, against RAD-500, and $4.3\times$ on KS, against the frozen default UNIF. On Burgers that method is frozen PINNACLE-K, within $1.3\times$ of SLED. On AC, KdV and RT the best established method (with or without refreshing) on each benchmark is within $1.3\times$ of SLED.

Frozen PINNACLE-K, the established method closest to SLED, selects from the tangent kernel after the warmup, but through the residual and without a Sobol component. Its median lies within $1.3\times$ of that of SLED on Burgers, $2\times$ and $7\times$ above it on AC and GL, and at least $9 \cdot 10^2\times$ above it on KdV, KS and RT. LEV lies at or below PINNACLE-K in the median on every benchmark.

Against its pure Sobol and pure leverage batches, SOBOL and LEV, the median of SLED lies at most $1.3\times$ above the better component on AC, GL, Burgers, KdV and RT, LEV on the first three and SOBOL on the last two. KS is the exception: the paired median against SOBOL is 1.37 ($[0.91, 2.60]$), and at $32{,}768$ it is 3.21 ($[2.17, 6.14]$; Appendix C.4). On RT, SLED is level with SOBOL and with the default in the median, with one failed seed of 10 (Appendix C.7). SLED selects the better component without knowing it in advance.

6 LIMITATIONS AND CONCLUSION

Limitations. We do not consider time-marching, and we test one architecture, periodic geometries and one second-phase optimizer, the dense self-scaled Broyden method, which is among the most accurate in recent comparisons (Urbán et al., 2025; Kiyani et al., 2025). Lemma 3.1 covers L-BFGS as well, but all reported results use the self-scaled Broyden method, and we claim nothing about the optimizer beyond the linearized model. The Sobol share of the Sobol–leverage design depends on Assumption 1. No benchmark studied here violates it, and Appendix A.7 describes the self-similar case that would. The leverage points of SLED are drawn from the Nyström estimate without a uniform mixture component, so the overestimate constant β of Theorem 4.2 is not guaranteed for them, and LEV and SLED draw with replacement, so the number of distinct points in a batch is below n (Appendix C.3). The theory is stated for the linearized second phase, and the network leaves it: at the end of the phase, $\varepsilon_{\mathrm{lin}}$ of eq. (2) is about ten times the residual after the warmup on KS and AC, the two benchmarks where we measured it, and the linearization predicts a ranking of the frozen batches, not their loss or error (Appendix D).

Conclusion. We asked which collocation points the quasi-Newton phase of two-phase PINN training should keep frozen. The batch fixes what the linearized phase returns; the network leaves that linearization, but the linearized fit still orders the frozen batches by their off-batch loss, and a batch is judged by that loss. SLED reached a median error at or below that of every established method on the six benchmarks, up to $60\times$ and $375\times$ below the frozen default, at 2.1–4.2 % of the second phase, and it avoided the failures of either of its components alone. A batch chosen once after the warmup matched or outperformed every refresh protocol tested. Two findings limit these answers: on Kuramoto–Sivashinsky a frozen Sobol set keeps the lower median $1.37\times$ at $16{,}384$ points, and the sample-complexity bound of Theorem 4.2 locates where leverage points alone fail without ordering the methods above it.

AI USE STATEMENT

In this work, we used generative AI tools (Anthropic Claude) for assisting in proofs, which the authors verified line by line. Furthermore it was used for writing verification scripts and parts of the experiment code, which the authors reviewed and tested. We have not used generative AI tools for generating synthetic data or for data cleaning, and the remaining required-disclosure tasks are not applicable. Additionally, we used generative AI tools for literature search and for drafting text that the authors then revised. All experiments were designed and run by the authors. All numerical results were produced by the authors' code and hardware. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work solves benchmark partial differential equations with synthetic reference solutions; it involves no human subjects, no personal data, and no application with a foreseeable harmful use.

REPRODUCIBILITY STATEMENT

The statements of the main text are proved in Appendix B. The supporting statements of Appendix A are proved where they are stated. Appendix C reports the pipeline configuration, precision and hardware, the reference solutions, the seed rosters and the results of every experiment, and Appendix D the linearization diagnostics with their provenance checks. Wall-clock is reported for the batch construction and the second phase of SLED and of the two refresh protocols (Appendix C.6). Batch rules are specified to the level of pseudocode (Section 4.2). The warmup checkpoints and configurations will be released with the code.

REFERENCES

Ben Adcock. Optimal sampling for least-squares approximation. *Foundations of Computational Mathematics*, 2025. arXiv:2409.02342.

Ben Adcock, Juan M. Cardenas, and Nick Dexter. CS4ML: A general framework for active learning with arbitrary data. In *NeurIPS*, 2023.

Nigel T. Andersen and Takashi Matsubara. PINNs failure modes are overfitting, 2026. arXiv:2605.30910.

Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *ICML*, 2017.

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

Albert Cohen and Giovanni Migliorati. Optimal weighted least-squares methods. *SMAI Journal of Computational Mathematics*, 3:181–203, 2017.

Albert Cohen, Mark A. Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of Computational Mathematics*, 13(5):819–834, 2013.

Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *SODA*, pp. 1758–1777, 2017.

Arka Daw, Jie Bu, Sifan Wang, Paris Perdikaris, and Anuj Karpatne. Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling. In *ICML*, volume 202 of *PMLR*, pp. 7264–7302, 2023.

Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *NeurIPS*, 2015.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *ICML*, volume 80 of *PMLR*, pp. 1832–1841, 2018.

Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

David Holzmüller, Viktor Zaverkin, Johannes Kästner, and Ingo Steinwart. A framework and benchmark for deep batch active learning for regression. *Journal of Machine Learning Research*, 24(164):1–81, 2023.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

Minseok Jeong, Giup Seo, and Euiseok Hwang. Mitigating instability in high residual adaptive sampling for PINNs via Langevin dynamics. In *NeurIPS*, 2025. Spotlight.

Anas Jnini, Elham Kiyani, Khemraj Shukla, Jorge F. Urbán, Nazanin Ahmadi Daryakenari, Johannes Müller, Marius Zeinhofer, and George Em Karniadakis. Curvature-aware optimization for high-accuracy physics-informed neural networks, 2026. Computer Methods in Applied Mechanics and Engineering; arXiv:2604.05230.

Elham Kiyani, Khemraj Shukla, Jorge F. Urbán, Jérôme Darbon, and George Em Karniadakis. Optimizing the optimizer for physics-informed neural networks and Kolmogorov–Arnold networks. *Computer Methods in Applied Mechanics and Engineering*, 446:118308, 2025. doi: 10.1016/j.cma.2025.118308. arXiv:2501.16371.

Gregory Kang Ruey Lau, Apivich Hemachandra, See-Kiong Ng, and Bryan Kian Hsiang Low. PINNACLE: PINN adaptive collocation and experimental points selection. In *ICLR*, 2024.

Jiale Linghu and Yangshuai Wang. Residual-Christoffel sampling for random feature collocation of linear PDEs, 2026. arXiv:2607.13382.

Marcello Longo, Siddhartha Mishra, T. Konstantin Rusch, and Christoph Schwab. Higher-order quasi-Monte Carlo training of deep neural networks. *SIAM Journal on Scientific Computing*, 43 (6):A3938–A3966, 2021.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

Siddhartha Mishra and T. Konstantin Rusch. Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences. *SIAM Journal on Numerical Analysis*, 59(3):1811–1834, 2021.

Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. In *NeurIPS*, 2017.

Francis J. Narcowich, Joseph D. Ward, and Holger Wendland. Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Mathematics of Computation*, 74(250):743–763, 2005.

Larry Nazareth. A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms. *SIAM Journal on Numerical Analysis*, 16(5):794–800, 1979.

Yurii Nesterov. *Lectures on Convex Optimization*. Springer, 2 edition, 2018.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

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

Pratik Rathore, Weimu Lei, Zachary Frangella, Lu Lu, and Madeleine Udell. Challenges in training PINNs: A loss landscape perspective. In *ICML*, 2024.

Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. In *NeurIPS*, 2018.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

Juan Diego Toscano, Zhaojie Chai, and George Em Karniadakis. GRAFT-ATHENA: Self-improving agentic teams for autonomous discovery and evolutionary numerical algorithms, 2026a. arXiv:2605.11117.

Juan Diego Toscano, Daniel T. Chen, Vivek Oommen, Jérôme Darbon, and George Em Karniadakis. A variational framework for residual-based adaptivity in neural PDE solvers and operator learning. *npj Artificial Intelligence*, 2(32), 2026b.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Jorge F. Urbán, Petros Stefanou, and José A. Pons. Unveiling the optimization process of physics informed neural networks: How accurate and competitive can PINNs be? *Journal of Computational Physics*, 523:113656, 2025. doi: 10.1016/j.jcp.2024.113656. arXiv:2405.04230.

Coen Visser, Alexander Heinlein, and Bianca Giovanardi. PACMANN: Point adaptive collocation method for artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 2025. arXiv:2411.19632.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam. In *ICLR*, 2025.

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

Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An expert's guide to training physics-informed neural networks. *arXiv preprint arXiv:2308.08468*, 2023.

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

Sifan Wang, Ananyae Kumar Bhartari, Bowen Li, and Paris Perdikaris. Gradient alignment in physics-informed neural networks: A second-order optimization perspective, 2025. arXiv:2502.00604.

Sifan Wang, Shawn Koohy, Yiping Lu, and Paris Perdikaris. When PINNs go wrong: Pseudo-time stepping against spurious solutions, 2026. arXiv:2604.23528.

Holger Wendland. *Scattered Data Approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2005.

Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *NeurIPS*, 2017.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

A SUPPORTING STATEMENTS FOR THE SAMPLING BOUND AND THE BATCH OF SLED

This appendix collects the statements that support the main text, grouped by what they are about. Every statement of this appendix that is not classical is proved or derived where it is stated. The three statements of the main text, Lemma 3.1, Theorem 4.2, and Proposition 4.4, are proved in Appendix B. Here and in Appendix B, the theory concerns the tangent features at the end of the warmup (Section 4.1). Only two of its statements bound the residual after the second phase: (a) the lower bound of Remark A.6 for a batch that misses a localized direction, and (b) the pointwise bound of Proposition 4.4(i) on the in-range part of the residual. The stability constant that relates the residual to the error of u_θ reported in Section 5 is given in Remark A.1; it applies to (a), and to (b) once the out-of-range part of the residual is also bounded.

A.1 WHAT THESE BOUNDS CONTROL

Remark A.1 (Stability constant). Every bound below controls the residual $\|r(\cdot\,;\theta)\|_\rho$, whereas the reported metric is the relative L^2 error of u_θ. For a linear operator $\mathcal{L}$ with bounded inverse, $\|u_\theta - u^\star\| \le \left\|\mathcal{L}^{-1}\right\| \, \|r(\cdot\,;\theta)\|$ up to boundary terms. For the nonlinear equations of Section 5 the same holds locally, under a local stability estimate for the linearized operator at $u^\star$ that we do not verify. The constant is a property of the PDE, not of the batch. A residual bound is therefore an error bound up to one constant of the PDE, the same for every batch rule; being one-sided, it does not by itself order the rules by error.

A.2 THE OBJECTS OF THE ANALYSIS

Spectral notation and conventions. Throughout this appendix and Appendix B, expectations without subscript are over $x \sim \rho$; sampling expectations are over $x_i \overset{\text{iid}}{\sim} q\,\mathrm{d}\rho$ with weights $\omega = 1/q$. Let

$$\Phi \;=\; \sum_{j=1}^{R} \sigma_j \, u_j \, v_j^\top, \qquad \sigma_1 \ge \cdots \ge \sigma_R > 0, \tag{4}$$

be the singular value decomposition of the feature operator, $\{v_j\}$ orthonormal in $\mathbb{R}^P$, $\{u_j\}$ orthonormal in $L^2(\rho)$, $\Phi v_j = \sigma_j u_j$, $\mathbf{G} v_j = \sigma_j^2 v_j$; wherever u_j is evaluated at a point we use the pointwise representative $u_j(x) = \varphi(x)^\top v_j/\sigma_j$. For ρ-a.e. x the feature vector lies in $\mathrm{range}(\mathbf{G})$, so that

$$\varphi(x) = \sum_j \sigma_j u_j(x)\, v_j, \qquad \kappa(x) = \sum_j \sigma_j^2 u_j(x)^2, \qquad \ell_\lambda(x) = \sum_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda}\, u_j(x)^2. \tag{5}$$

These *spectral identities* follow from (4) once $\varphi(x) \in \mathrm{range}(\mathbf{G})$ for ρ-a.e. x: if $v \perp \mathrm{range}(\mathbf{G})$ then $0 = v^\top \mathbf{G} v = \mathbb{E}[(\varphi^\top v)^2]$, so $\varphi(x)^\top v = 0$ for ρ-a.e. x; apply this to a finite basis of $\mathrm{range}(\mathbf{G})^\perp$, expand $\varphi(x)$ in the basis $\{v_j\}$ of $\mathrm{range}(\mathbf{G})$, and use $\varphi(x)^\top v_j = (\Phi v_j)(x) = \sigma_j u_j(x)$.

Remark A.2 (Pool discretization). Implementations draw a pool $S = \{z_1, \ldots, z_M\} \overset{\text{iid}}{\sim} \rho$, form the pmf $q_j \propto s(z_j)$ over S from the score s, sample indices with replacement, and use exact weights $\omega_j = 1/(M q_j)$. This is i.i.d. sampling from the density q conditionally on S, and all statements hold conditionally on the pool with ρ replaced by the empirical measure of S; the outer $M^{-1/2}$ pool fluctuation is standard Monte Carlo and independent of the batch rule. Every population quantity the paper reports (ℓ_λ, d_λ, $\sup \ell_\lambda$, $\widehat{\rho}$) is therefore a pool quantity: $\widehat{d_\lambda}$ is a Monte Carlo estimate, and $\sup_S \widehat{\ell_\lambda}$ estimates $\sup_D \ell_\lambda$, so the reported coherence is an estimate of the population coherence. For the population leverage at the pool points, $\max_S \ell_\lambda \le \sup_D \ell_\lambda$; the estimate $\widehat{\ell_\lambda}$ is computed under the empirical measure of the pool and can exceed $\sup_D \ell_\lambda$ where the pool under-samples a direction.

Remark A.3 (Multiple residual families). With residuals $\{r_e\}_{e \in \mathcal{E}}$ (PDE, boundary, initial), each family carries its own point set and features $\varphi_e = \nabla_\theta r_e(\cdot\,;\theta_0)$; the analysis applies per family, with $V_C = \mathrm{span} \bigcup_e \{\varphi_e(x_i^e)\}$ in Lemma 3.1, and the rows of every family count in Lemma 3.1(iii). Outer task weights multiply (1) by constants and commute with all statements. Which families the experiments carry, which of them is sampled, and the grid the initial condition is evaluated on are specified in Appendix C.

A.3 THE KERNEL DIAGONAL AND THE LEVERAGE FUNCTION

Proposition A.4 (κ versus ℓ_λ)**.** *For ρ-a.e. x:* (i) (two-sided bound)

$$\frac{\kappa(x)}{\lambda_{\max}(\mathbf{G})+\lambda} \;\le\; \ell_\lambda(x) \;\le\; \frac{\kappa(x)}{\lambda^+_{\min}(\mathbf{G})+\lambda},$$

so $q_\kappa := \kappa/\operatorname{tr}\mathbf{G}$ is a β_κ-leverage-overestimate with $\beta_\kappa \ge d_\lambda\big(\lambda^+_{\min}(\mathbf{G})+\lambda\big)/\operatorname{tr}\mathbf{G}$, and $\beta_\kappa = 1$ when the nonzero spectrum of $\mathbf{G}$ is flat. (ii) (the regularization path) *$\lambda \mapsto \ell_\lambda(x)$ is non-increasing, $\lambda \mapsto \lambda\ell_\lambda(x)$ is non-decreasing, and*

$$\lim_{\lambda\to\infty} \lambda\,\ell_\lambda(x) = \kappa(x), \qquad \lim_{\lambda\to 0} \ell_\lambda(x) = \ell_0(x) := \varphi(x)^\top \mathbf{G}^+ \varphi(x) = \sum_{j\le R} u_j(x)^2.$$

(iii) (rank correlation with κ) *On any pool without ties in κ on which ℓ_λ is not constant, and at any λ,* $\operatorname{Spearman}(\ell_\lambda, \kappa) = \lim_{\lambda'\to\infty} \operatorname{Spearman}(\ell_\lambda, \ell_{\lambda'})$.

Proof. Part (i). By (5), pointwise

$$\frac{\ell_\lambda(x)}{\kappa(x)} = \frac{\sum_j \frac{\sigma_j^2}{\sigma_j^2+\lambda} u_j(x)^2}{\sum_j \sigma_j^2 u_j(x)^2}$$

is a convex combination of the values $\{(\sigma_j^2+\lambda)^{-1}\}$ with weights $\sigma_j^2 u_j(x)^2/\kappa(x)$, hence lies in $[(\sigma_1^2+\lambda)^{-1}, (\sigma_R^2+\lambda)^{-1}] = [(\lambda_{\max}(\mathbf{G})+\lambda)^{-1}, (\lambda^+_{\min}(\mathbf{G})+\lambda)^{-1}]$, which is the two-sided bound. For the overestimate constant, with $q_\kappa = \kappa/\mathbb{E}[\kappa] = \kappa/\operatorname{tr}\mathbf{G}$,

$$\frac{q_\kappa(x)}{\ell_\lambda(x)/d_\lambda} = \frac{\kappa(x)}{\ell_\lambda(x)} \cdot \frac{d_\lambda}{\operatorname{tr}\mathbf{G}} \;\ge\; \big(\lambda^+_{\min}(\mathbf{G})+\lambda\big)\frac{d_\lambda}{\operatorname{tr}\mathbf{G}} \;=:\; \beta_\kappa.$$

Flat spectrum $\sigma_j \equiv \sigma$: $\ell_\lambda = \kappa/(\sigma^2+\lambda)$, $d_\lambda = R\sigma^2/(\sigma^2+\lambda)$, $\operatorname{tr}\mathbf{G} = R\sigma^2$, so $\beta_\kappa = 1$ and $q_\kappa = \ell_\lambda/d_\lambda$ exactly.

Parts (ii) and (iii). By (5), $\lambda\ell_\lambda = \sum_j \sigma_j^2 u_j^2 \cdot \lambda/(\sigma_j^2+\lambda)$, and $\lambda/(\sigma_j^2+\lambda)$ increases to 1 as $\lambda\to\infty$; the limit is κ, and each term increases in λ. As $\lambda \downarrow 0$ every weight $\sigma_j^2/(\sigma_j^2+\lambda)$ increases to 1, so ℓ_λ decreases in λ and tends to $\sum_{j\le R} u_j^2$, which equals $\varphi^\top\mathbf{G}^+\varphi$ because $\mathbf{G}^+ = \sum_j \sigma_j^{-2} v_j v_j^\top$ and $\varphi^\top v_j = \sigma_j u_j$. A rank correlation is invariant under strictly increasing transformations of either argument, in particular under scaling by $\lambda' > 0$, so $\operatorname{Spearman}(\ell_\lambda, \ell_{\lambda'}) = \operatorname{Spearman}(\ell_\lambda, \lambda'\ell_{\lambda'})$, and on a finite pool the right side is eventually constant in λ' and equal to $\operatorname{Spearman}(\ell_\lambda, \kappa)$ (for a pool without ties in κ), since $\lambda'\ell_{\lambda'} \to \kappa$ pointwise and the pool is finite. □

By part (i), q_κ is a β_κ-leverage-overestimate, so Theorem 4.2 applies to a batch drawn from it. However, the lower bound on β_κ equals 1 only on a flat nonzero spectrum. On a steep spectrum, $\lambda^+_{\min}(\mathbf{G})$ tends to 0 and $d_\lambda \ll R$, while the largest eigenvalues make up most of $\operatorname{tr}\mathbf{G}$. The lower bound then tends to $d_\lambda\lambda/\operatorname{tr}\mathbf{G}$, which is at most $\lambda_{\mathrm{rel}} d_\lambda$ at $\lambda = \lambda_{\mathrm{rel}}\lambda_{\max}(\mathbf{G})$, the population form of the ridge of Algorithm 1. The prefactor d_λ/β_κ of the batch-size condition of Theorem 4.2 for q_κ is then bounded only by $\operatorname{tr}\mathbf{G}/\big(\lambda^+_{\min}(\mathbf{G})+\lambda\big)$, which tends to $\operatorname{tr}\mathbf{G}/\lambda \ge \lambda_{\mathrm{rel}}^{-1}$. With $\varepsilon = \frac{1}{2}$ and $\lambda^+_{\min}(\mathbf{G}) \le \lambda$, the batch size that Theorem 4.2 requires under this bound exceeds $16\lambda_{\mathrm{rel}}^{-1}$, since $\log(16 d_\lambda/\zeta) > 2$, and lies above the operating batch size $n = 16{,}384$. By contrast, the prefactor for the leverage density itself is d_λ, which lies below this bound, since $d_\lambda = \sum_j \sigma_j^2/(\sigma_j^2+\lambda) \le \operatorname{tr}\mathbf{G}/\big(\lambda^+_{\min}(\mathbf{G})+\lambda\big)$. For this reason, Algorithm 1 draws its leverage points from $\widehat{\ell}_\lambda$ and not from $\widehat{\kappa}$ (Section 4.1).

Parts (ii) and (iii) describe how ℓ_λ depends on the ridge λ, that is, the *regularization path* $\lambda \mapsto \ell_\lambda$ of part (ii). By part (ii), the ranking of a pool by ℓ_λ tends to the ranking by the kernel diagonal κ as $\lambda\to\infty$, since $\lambda\ell_\lambda \to \kappa$, and to the ranking by ℓ_0 as $\lambda \to 0$, since $\ell_\lambda \to \ell_0$. Since $\mathbb{E}[\ell_0] = R = \dim\operatorname{range}(\mathbf{G})$, the density ℓ_0/R is the *optimal sampling density* of Cohen & Migliorati (2017) for least squares on the range of Φ. As $\lambda \to 0$, which corresponds to the interpolating second phase (Remark A.5), the leverage density of Definition 4.1 tends to ℓ_0/R, so a leverage batch for that phase would be drawn from the optimal sampling density ℓ_0/R. However, every singular function u_j

with $j \le R$ enters $\ell_0 = \sum_{j \le R} u_j^2$ with coefficient 1, independently of σ_j. The leverage of a Nyström approximation of rank k therefore coincides with ℓ_0 only if $R \le k$, where $k = 256$ in Algorithm 1. Since the only available bound on R, namely $R \le P$, is far above k (Table 5), Algorithm 1 does not attempt to estimate ℓ_0. Instead, it computes $\widehat{\rho} = \max\{0, \mathrm{Spearman}(\widehat{\ell_\lambda}, \widehat{\kappa})\}$ on the pool. By part (iii), $\widehat{\rho}$ is, up to the estimates $\widehat{\ell_\lambda}$ and $\widehat{\kappa}$, the clipping at 0, and ties, the rank correlation between the leverage rankings of the pool at the ridge λ and in the limit $\lambda \to \infty$. The rank correlation $\widehat{\rho}$ thus measures one quantity, the agreement between the leverage rankings of the pool at these two points of the regularization path.

A.4 Ridge and interpolation

Remark A.5 (The ridge is an analysis device). Theorem 4.2 is stated at a ridge $\lambda > 0$. For a fixed batch and weights with $\{\varphi(x_i)\}$ linearly independent, the weighted ridge solution $\widehat{\delta}_\lambda := \arg\min_\delta \widehat{L}(\delta) + \lambda \|\delta\|_2^2$ tends to the minimum-norm interpolant δ^{mn} of Lemma 3.1 as $\lambda \to 0^+$, and the weights cancel in the limit. This is standard, through the SVD of $W^{1/2}\Phi_C$: the minimum-norm solution of a consistent system depends only on its solution set and the row space V_C. The interpolating phase is therefore the limit $\lambda \to 0$ of the family the theorem describes. The theorem's constants degrade as $\lambda \to 0$ ($d_\lambda \uparrow R$, β_κ shrinks with $\lambda^+_{\min} + \lambda$). We therefore state the theory at a fixed ridge λ matched to the resolvable spectrum ($\lambda_{\mathrm{rel}} = 10^{-4}$, Algorithm 1) and claim no ridgeless bound.

A.5 When leverage helps, and when it does not

Recall that the coherence $\sup \ell_\lambda / d_\lambda$ is the ratio of the prefactor of the uniform batch-size condition of Theorem 4.2 to that of the leverage one (Section 4.1). The following two remarks describe one mechanism that makes the coherence large and one from which a coherence close to 1 follows. First, let a singular function u_j be supported on a set A of ρ-measure η. Then $\int_A \ell_\lambda \, \mathrm{d}\rho \ge \sigma_j^2/(\sigma_j^2 + \lambda)$, so $\sup \ell_\lambda \ge \sigma_j^2/\big(\eta(\sigma_j^2 + \lambda)\big)$, and the coherence is large when ηd_λ is small. A uniform batch misses A with probability $(1-\eta)^n$ (Remark A.6). Second, let J be an equidistributed set of singular indices (Remark A.7). Then the terms with $j \in J$ contribute a constant to ℓ_λ and a constant to κ, so at every λ both rankings over any pool are those of the terms outside J. By Remark A.7, $\sup \ell_\lambda = \sup \ell_\lambda^{J^c} + d_\lambda^J$ and $d_\lambda \ge d_\lambda^J$, so the coherence is at most $1 + \sup \ell_\lambda^{J^c}/d_\lambda^J$. In particular, it is close to 1 when $\sup \ell_\lambda^{J^c} \ll d_\lambda^J$, and the two sample-complexity bounds of Theorem 4.2 then nearly coincide (Section 4.1). If, moreover, $\sup \ell_0^{J^c} \ll R$, the optimal sampling density ℓ_0/R is nearly uniform and the fill distance is the only quantity left to optimize (Remark A.7(ii)).

The following remark is a model statement: for the real-analytic networks of Section 5 the support condition $\mathrm{supp}(u_j) \subseteq A$ holds only approximately, and the direction v_j is then only nearly orthogonal to V_C.

Remark A.6 (Localized directions are missed by density-blind batches). Let (σ_j, u_j, v_j) be the SVD (4) and suppose some singular function is localized: there are j and $A \subset D$ with $\rho(A) = \eta$ and $\mathrm{supp}(u_j) \subseteq A$; let $c_j := \langle r_0, u_j \rangle_\rho$. If no batch point lies in A, which for the uniform rule is the event $C \cap A = \emptyset$ of probability $(1-\eta)^n$, then every method satisfying the linear-span assumption (Lemma 3.1) returns δ_∞ with $v_j^\top \delta_\infty = 0$, hence $\|r_0 + \Phi\delta_\infty\|_\rho^2 \ge c_j^2$, and for the uniform rule $\mathbb{E}\big[\|r_0 + \Phi\delta_\infty\|_\rho^2\big] \ge (1-\eta)^n c_j^2$. The leverage mass of A is at least $\sigma_j^2/\big(d_\lambda(\sigma_j^2 + \lambda)\big)$, so a β-leverage-overestimate places at least $\beta n \, \sigma_j^2/\big(d_\lambda(\sigma_j^2 + \lambda)\big)$ points in A in expectation, independently of η. This is the mechanism behind the $\sup \ell_\lambda$ of the uniform condition (a single localized direction forces $n \gtrsim 1/\eta$) and behind the draw-to-draw variance of the uniform batches in Figure 1.

Proof. With u_j taken pointwise, $u_j(x_i) = 0$ for every batch point $x_i \notin A$, hence $\varphi(x_i)^\top v_j = \sigma_j u_j(x_i) = 0$ and $v_j \perp V_C$. Such a method returns $\delta_\infty \in V_C$ (Lemma 3.1), so $v_j^\top \delta_\infty = 0$ and $\langle r_0 + \Phi\delta_\infty, u_j \rangle_\rho = c_j + \sigma_j v_j^\top \delta_\infty = c_j$, using $\langle \Phi\delta, u_j \rangle = \langle \delta, \Phi^* u_j \rangle = \sigma_j v_j^\top \delta$; Bessel's inequality gives $\|r_0 + \Phi\delta_\infty\|_\rho^2 \ge c_j^2$, and multiplying by the miss probability $(1-\eta)^n$ of the uniform rule gives the expectation bound. For the leverage mass, by (5) and $\mathrm{supp}(u_j) \subseteq A$,

$\int_A \ell_\lambda \,\mathrm{d}\rho \geq \frac{\sigma_j^2}{\sigma_j^2+\lambda} \int_A u_j^2 \,\mathrm{d}\rho = \frac{\sigma_j^2}{\sigma_j^2+\lambda}$, so a β-overestimate places at least $\beta n\, \sigma_j^2/(d_\lambda(\sigma_j^2+\lambda))$ points in A in expectation. □

Remark A.7 (Equidistributed index sets are invisible to the ranking). Call a set J of singular indices *equidistributed* if for every singular value σ occurring in J the sum $\sum_{j\in J,\,\sigma_j=\sigma} u_j(x)^2$ is ρ-a.e. constant (a Fourier band on a torus, or any spectral band of a translation-invariant kernel). Then for every $\lambda \in (0,\infty]$

$$\ell_\lambda = \ell_\lambda^{J^c} + d_\lambda^J, \qquad \kappa = \kappa^{J^c} + \operatorname{tr}_J \mathbf{G},$$

with $\ell_\lambda^{J^c} = \sum_{j\notin J} w_j(\lambda) u_j^2$, $w_j(\lambda) = \sigma_j^2/(\sigma_j^2+\lambda)$, and $d_\lambda^J = \sum_{j\in J} w_j(\lambda)$: the set J adds a constant to both densities. Hence *(i)* the ranking of ℓ_λ over any pool, at every λ, is that of the complement J^c; and *(ii)* if $|J^c| \ll R$ the optimal sampling density ℓ_0/R is uniform up to the factor $1+\sup \ell_0^{J^c}/R$, and the point set with the smallest fill-distance bound on $\sup P_C$ is the low-discrepancy set (Lemma A.9).

Proof. Group J by singular value. Within a group every $w_j(\lambda)$ is the same number $w(\lambda)$, so $\sum_{\text{group}} w_j u_j^2 = w(\lambda)\sum_{\text{group}} u_j^2 = w(\lambda)\cdot\text{const}$; summing over the groups gives $\ell_\lambda = \ell_\lambda^{J^c} + \text{const}(\lambda)$, and the constant is $\sum_{j\in J} w_j(\lambda) = d_\lambda^J$ because each u_j has unit $L^2(\rho)$ norm. The same computation with σ_j^2 in place of w_j gives $\kappa = \kappa^{J^c} + \operatorname{tr}_J \mathbf{G}$. Adding a constant preserves every ranking, which is (i). For (ii), as $\lambda \to 0$ the weights tend to 1, $d_\lambda^J \to |J|$, $d_\lambda^{J^c} \to |J^c|$, so $\sup \ell_0/\mathbb{E}\ell_0 \leq (\sup \ell_0^{J^c} + |J|)/R \leq 1 + \sup \ell_0^{J^c}/R$; a density that is uniform to a constant factor has, for an i.i.d. draw, the fill distance of an i.i.d. uniform draw up to that factor, which Lemma A.9 puts above that of the low-discrepancy set by the logarithm. □

A.6 Scope of Theorem 4.2 in the batch size

Remark A.8 (Scope in n). Theorem 4.2 constrains the batch size, and what it requires of n differs from pool to pool. Without the theorem's constants the sample-complexity bound is $d_\lambda \log d_\lambda$ for a leverage batch and $\sup \ell_\lambda$ for a uniform one. With the constants of Section 4.1 it is the bound n_{Lev} displayed there, with the pool estimate $\widehat{d_\lambda}$ in place of d_λ, and the uniform bound, with $\sup_S \widehat{\ell_\lambda}$ in place of the prefactor $\widehat{d_\lambda}$. Both are estimated on the 65,536-point pools after the warmup ($\lambda_{\text{rel}} = 10^{-4}$, seed medians), and both are computed on the plain pool of Lev, which is the pool the theorem's leverage batch is drawn from; the pool of the Sobol–leverage design is the same pool, and its Sobol component is not part of it. Both are pool quantities in the sense of Remark A.2: $\max_S \ell_\lambda \leq \sup_D \ell_\lambda$ for the population leverage, whereas the estimate $\sup_S \widehat{\ell_\lambda}$ can lie on either side of $\sup_D \ell_\lambda$. Two consequences bound what the theorem can be asked to do. First, on a pool where both bounds lie below n, the theorem guarantees the same ε-embedding for a leverage batch and a uniform batch, each with probability at least $1-\zeta$, and therefore cannot order them: the sampling statements are *saturated* there, and they are not what separates the methods of Table 1. Second, the low-discrepancy batch is not an i.i.d. draw and is outside the theorem's hypotheses altogether; what covers it is the fill distance of Lemma A.9 and Corollary A.10. The bounds per benchmark, the point of diminishing returns of each method (Section 5.1), and the comparison of the two are reported in Section 5.3 and Appendix C.4.

A.7 The Sobol share of the Sobol–leverage design

This subsection collects what the theory fixes about the Sobol share μ_{sob} of Algorithm 1, and what it leaves open. Recall that P_C denotes the power function of Proposition 4.4 (Section 4.2). We write Π_C for the orthogonal projector onto the row space V_C, as in the proof of that proposition in Appendix B.

Proposition 4.4(ii) is proved in Appendix B from the monotonicity of Π_C under inclusion of batches. In particular, the bound on the union is deterministic and needs no i.i.d. assumption, so it applies although the Sobol component is not an i.i.d. sample, and its bound $P_C \leq \min\{P_{C_{\text{cov}}}, P_{C_{\text{lev}}}\}$ has constant 1. Through it, the bound of Corollary A.10 on the Sobol component holds for the whole batch. The bound on the union holds with equality almost everywhere if and only if the row space of one part of the batch contains that of the other (Remark A.12).

Lemma A.9 (Fill distance; classical)**.** *Let $h_C := \sup_{x\in D} \min_{x_i\in C} |x - x_i|$. For a kernel of smoothness order r on a bounded Lipschitz domain $D \subset \mathbb{R}^d$ there are c_K and h_0, depending on K,*

D, d, and r and not on C, with $\sup_x P_C(x) \le c_K\, h_C^r$ for $h_C \le h_0$ (Wendland, 2005, Ch. 11); see also Narcowich et al. (2005). A scrambled Sobol set of m points has $h \asymp m^{-1/d}$, which is optimal; m i.i.d. uniform points have $h \asymp (\log m/m)^{1/d}$; and m i.i.d. points from a density q have, on a region A, a fill distance governed by $q(A)\, m$. The fill distance is a property of the point set alone: it does not depend on the features.

Corollary A.10 (Fill-distance bound of the Sobol component)**.** *Let the Sobol component of Algorithm 1 have $m = \mu_{\mathrm{sob}} n$ points. Then $\sup_x P_{C_{\mathrm{cov}}}(x) \le c'_K\, (\mu_{\mathrm{sob}} n)^{-r/d}$, so that the bound of Lemma A.9 for a Sobol component of share μ_{sob} exceeds the bound for the Sobol set of all n points by the factor $\mu_{\mathrm{sob}}^{-\varsigma}$, $\varsigma := r/d$.*

Proof. Lemma A.9 with $h \asymp m^{-1/d}$ and $m = \mu_{\mathrm{sob}} n$; the ratio of the two bounds at $m = \mu_{\mathrm{sob}} n$ and $m = n$ is $\mu_{\mathrm{sob}}^{-r/d}$. □

Lemma A.9 is stated for kernels of finite smoothness order, and the empirical tangent kernel has finite rank, so the exponent ς is not fixed for the networks of Section 5, and Corollary A.10 does not predict the error of a Sobol component.

Corollary A.11 (Below the critical share, the batch keeps the embedding)**.** *Let the leverage points C_{lev} of Algorithm 1 consist of $n_{\mathrm{lev}} \ge (1-\mu_{\mathrm{sob}})n$ i.i.d. draws from the leverage density ($\beta = 1$; the Nyström estimate enters through β), let n_{Lev} be the batch size that Theorem 4.2 requires at $\beta = 1$, the sample-complexity bound of Section 4.1, and let $\widehat{\mathbf{G}}_C$ be the weighted Gram of the batch $C = C_{\mathrm{cov}} \cup C_{\mathrm{lev}}$ (weights 1 on the Sobol points, $1/(Mq_i)$ on the leverage points). Under the conditions of Theorem 4.2, if $(1-\mu_{\mathrm{sob}})n \ge n_{\mathrm{Lev}}$, then with probability at least $1-\zeta$,*

$$\widehat{\mathbf{G}}_C + \lambda I \;\succeq\; (1-\mu_{\mathrm{sob}})(1-\varepsilon)\,(\mathbf{G}+\lambda I) \qquad \textit{and} \qquad P_C \le P_{C_{\mathrm{lev}}} \textit{ pointwise :}$$

the batch keeps the lower embedding bound of its leverage points, up to the factor $1-\mu_{\mathrm{sob}}$, and their power function, for every $\mu_{\mathrm{sob}} \le \mu_{\mathrm{crit}} := 1 - n_{\mathrm{Lev}}/n$, the critical share*. Above μ_{crit} the leverage points lie below their sample-complexity bound, and the batch is covered only by the bound of its Sobol component (Corollary A.10).*

Proof. Let $n_{\mathrm{lev}} \ge (1-\mu_{\mathrm{sob}})n$ be the number of leverage points and let $\widehat{\mathbf{G}}_{\mathrm{lev}} := \frac{1}{n_{\mathrm{lev}}} \sum_{i \in C_{\mathrm{lev}}} \omega_i \varphi(x_i)\varphi(x_i)^\top$ be the weighted Gram of the leverage points alone, the estimator of Theorem 4.2 with n_{lev} draws. Since $n_{\mathrm{lev}} \ge n_{\mathrm{Lev}}$, the theorem gives $\widehat{\mathbf{G}}_{\mathrm{lev}} + \lambda I \succeq (1-\varepsilon)(\mathbf{G}+\lambda I)$ with probability at least $1-\zeta$. The Gram of the batch is $\widehat{\mathbf{G}}_C = \frac{1}{n}\sum_{i \in C_{\mathrm{cov}}} \varphi(x_i)\varphi(x_i)^\top + \frac{n_{\mathrm{lev}}}{n}\, \widehat{\mathbf{G}}_{\mathrm{lev}} \succeq (1-\mu_{\mathrm{sob}})\, \widehat{\mathbf{G}}_{\mathrm{lev}}$, the Sobol term being positive semidefinite, hence $\widehat{\mathbf{G}}_C + \lambda I \succeq (1-\mu_{\mathrm{sob}})(\widehat{\mathbf{G}}_{\mathrm{lev}} + \lambda I) \succeq (1-\mu_{\mathrm{sob}})(1-\varepsilon)(\mathbf{G}+\lambda I)$ on that event. The power-function inequality is Proposition 4.4(ii) with $C_{\mathrm{lev}} \subseteq C$, and holds for every draw. □

Corollary A.11 applies with $\beta = 1$ when $\mu_{\mathrm{sob}} \le \mu_{\mathrm{crit}}$; when $\mu_{\mathrm{sob}} > \mu_{\mathrm{crit}}$, the batch of the Sobol–leverage design is covered only by the bound of its Sobol component. At $n = 16{,}384$ the critical share is positive, $\mu_{\mathrm{crit}} > 0$, on four of the six benchmarks of Section 5 (Burgers, Allen–Cahn, Ginzburg–Landau, and KdV). At the medians over seeds of n_{Lev} in Table 8, computed on the plain pool of LEV (Remark A.8), the critical share is 0.73, 0.66, and 0.54 on Burgers, Allen–Cahn, and Ginzburg–Landau, respectively. Since $\mu_{\mathrm{sob}} = \widehat{\rho}$ and the seed medians 0.58, 0.00, and 0.40 of $\widehat{\rho}$ lie below the respective critical shares, Corollary A.11 applies on these three benchmarks at the median values of $\widehat{\rho}$. The highest seed value of $\widehat{\rho}$ on Burgers, the upper end of 0.48–0.74, lies above the median-based critical share, so the corollary does not apply to that seed. The highest seed value on Ginzburg–Landau, the upper end of 0.33–0.54, lies at its critical share to two digits. On KdV, the critical share is 0.33 and the median 0.88 lies above it; on Kuramoto–Sivashinsky and Rayleigh–Taylor, $n_{\mathrm{Lev}} > n$, so $\mu_{\mathrm{crit}} < 0$. On these three benchmarks, the power function of the batch of the Sobol–leverage design is therefore bounded only through Proposition 4.4(ii) together with Corollary A.10.

Note that every critical share of the six benchmarks is at most 0.73, so Corollary A.11 does not apply whenever $\widehat{\rho} > 0.73$, in particular whenever $\widehat{\rho}$ is near 1. For a value of $\widehat{\rho}$ well below 1, it applies only if that value also lies at or below the critical share of its benchmark. Recall from the discussion after Proposition A.4 that $\widehat{\rho}$ measures the agreement between the leverage rankings of the pool at the

ridge λ and in the limit $\lambda \to \infty$, and from Section 4.2 that agreement calls for a space-filling design and disagreement for leverage points. Since the interpolating second phase corresponds to the limit $\lambda \to 0$ (Remark A.5), the frozen batch is used at ridge values below λ, at which $\widehat{\rho}$ does not compare the rankings. Section 4.2 therefore rests on the following assumption.

Assumption 1 (Stable leverage ranking)**.** A leverage ranking of the pool that is stable over the ridge values resolved at the end of the warmup stays stable for ridge values below them.

We do not prove Assumption 1. It fails on a self-similar problem whose singular functions with $\sigma_j^2 \ll \lambda$ are concentrated where those with $\sigma_j^2 \gtrsim \lambda$ are, and are more localized. On such a problem, a value of $\widehat{\rho}$ near 1 would give the Sobol component almost all of the batch, although more leverage points would lower the error. No benchmark studied here shows evidence of such a case. On KdV, Kuramoto–Sivashinsky, and Rayleigh–Taylor, where $\widehat{\rho}$ exceeds the critical share, the paired median error ratio of the Sobol–leverage design to the Sobol set at $n = 16{,}384$ is 0.96×, 1.37×, and 1.18×, respectively, and every paired interval contains 1 (Table 1).

The identity $\mu_{\mathrm{sob}} = \widehat{\rho}$ is one choice among maps $\mu_{\mathrm{sob}} = f(\widehat{\rho})$, and two statements of this subsection constrain any such map. First, by Corollary A.10, the bound on the power function of a Sobol component of share μ_{sob} is $\mu_{\mathrm{sob}}^{-r/d}$ times that of a Sobol set of all n points, so the bound grows as μ_{sob} decreases. Second, the embedding bound of Corollary A.11 is proved only for $\mu_{\mathrm{sob}} \le \mu_{\mathrm{crit}}$, so Corollary A.11 applies to a map only if $f(\widehat{\rho}) \le \mu_{\mathrm{crit}}$. By the discussion after Corollary A.11, the identity map satisfies $\mu_{\mathrm{sob}} \le \mu_{\mathrm{crit}}$ at the seed medians of $\widehat{\rho}$ on Burgers, Allen–Cahn, and Ginzburg–Landau, and violates it on KdV, Kuramoto–Sivashinsky, and Rayleigh–Taylor; on the last two, where $\mu_{\mathrm{crit}} < 0$, Corollary A.11 applies to no map, and Corollary A.10 is the only constraint. A map with $f \ge \mathrm{id}$ on $[0, 1]$ satisfies $f(\widehat{\rho}) \le \mu_{\mathrm{crit}}$ on at most the benchmarks on which the identity map does; for example, $\mu_{\mathrm{sob}} = \sqrt{\widehat{\rho}}$ violates $\mu_{\mathrm{sob}} \le \mu_{\mathrm{crit}}$ at the seed medians also on Burgers and Ginzburg–Landau.

Remark A.12 (When the minimum in Proposition 4.4(ii) is attained). Equality at a single point x can be characterized. Since $V_{C_i} \subseteq V_C$ for $C_i \in \{C_{\mathrm{cov}}, C_{\mathrm{lev}}\}$, the identity $P_C^2 = \|(I - \Pi_C)\varphi\|^2$ of Appendix B gives $P_{C_i}(x)^2 = P_C(x)^2 + \|(\Pi_C - \Pi_{C_i})\varphi(x)\|^2$. Equality therefore holds at x if and only if $\varphi(x)$ is orthogonal to the orthogonal complement $W_i := V_C \ominus V_{C_i}$ of V_{C_i} in V_C for a part C_i that attains the minimum at x. For the networks of Section 5 the features are real-analytic on the connected domain D, and ρ is the uniform measure. If $W_i \neq 0$, the analytic map $x \mapsto (\Pi_C - \Pi_{C_i})\varphi(x)$ does not vanish identically, since otherwise $W_i \perp \mathrm{span}\{\varphi(x) : x \in D\} \supseteq W_i$; its zero set therefore has ρ-measure zero. Hence equality holds ρ-a.e. if and only if $W_{\mathrm{cov}} = 0$ or $W_{\mathrm{lev}} = 0$, that is, if and only if the row space of one part of the batch contains that of the other. For the batch of SLED, whose two parts have different row spaces, the minimum is therefore not attained on a set of positive measure, and the bound of part (ii) is strict almost everywhere.

B PROOFS OF THE STATEMENTS OF THE MAIN TEXT

In this appendix, we prove the four statements of the main text, namely Lemma 3.1, Theorem 4.2, Corollary 4.3, and Proposition 4.4. The supporting statements are proved or derived in Appendix A, where they are stated.

Lemma 3.1 has classical antecedents: Wilson et al. (2017) and Gunasekar et al. (2018) for the confinement of the iterates to the row span and the minimum-norm limit of gradient methods, adaptive preconditioners excluded, and Nazareth (1979) for BFGS with exact line searches, by its equivalence with conjugate gradients. The updated residual field of part (ii) is the linearized-training statement of Jacot et al. (2018) in the residual form of Wang et al. (2022), and the collocation solution of Chen et al. (2021) with the empirical tangent kernel. Our part is the class of part (i), the Broyden family with a scalar initial matrix, any line search and the self-scaled variants included, and the weighted, inconsistent case of part (iii).

Throughout, the notation and the conventions on expectations are those of Appendix A.

B.1 Proof of Lemma 3.1(i)

Proof. For the linear model $r_0 + \Phi\delta$ of eq. (2), the objective of the second phase is $\widehat{L}(\delta) = \frac{1}{n}\sum_{i=1}^{n} \omega_i\big(r_0(x_i) + \varphi(x_i)^\top\delta\big)^2$. Its gradient is

$$\nabla\widehat{L}(\delta) = \frac{2}{n}\sum_{i=1}^{n} \omega_i\big(r_0(x_i) + \varphi(x_i)^\top\delta\big)\varphi(x_i) \;\in\; V_C \qquad \text{for every } \delta. \tag{6}$$

We first show by induction on t that every method satisfying the linear-span assumption has $\delta_t \in V_C$. For $t = 0$ we have $\delta_0 = 0 \in V_C$. Assume that $\delta_0, \dots, \delta_t \in V_C$. Then $g_0, \dots, g_t \in V_C$ by (6). Since the method satisfies the linear-span assumption, δ_{t+1} lies in $\operatorname{span}(\{\delta_0, \dots, \delta_t\} \cup \{g_0, \dots, g_t\})$, and this span is contained in V_C.

Next, we show that the first-order methods listed in Lemma 3.1(i) satisfy the linear-span assumption. We use the convention $\delta_{-1} := \delta_0$, so that $g_{-1} = g_0$. For gradient descent, $\delta_{t+1} = \delta_t - \alpha_t g_t$. For the momentum method, $\delta_{t+1} = \delta_t - \alpha_t g_t + \mu(\delta_t - \delta_{t-1})$. For Nesterov's method, $\delta_{t+1} = \delta_t + \mu(\delta_t - \delta_{t-1}) - \alpha_t\nabla\widehat{L}(\delta_t + \mu(\delta_t - \delta_{t-1}))$, and the gradient in this update equals $(1+\mu)g_t - \mu g_{t-1}$, because $\nabla\widehat{L}$ is affine. For nonlinear conjugate gradients, $\delta_{t+1} = \delta_t + \alpha_t p_t$ with $p_0 = -g_0$ and $p_t = -g_t + \beta_t p_{t-1}$, so that $p_t \in \operatorname{span}\{g_0, \dots, g_t\}$ by induction on t. In each case, δ_{t+1} lies in $\operatorname{span}(\{\delta_0, \dots, \delta_t\} \cup \{g_0, \dots, g_t\})$.

We now treat the Broyden class with $H_0 = \gamma I$. For BFGS with $H_0 = I$ and exact line searches on a quadratic, it is classical that the iterates coincide with those of conjugate gradients and lie in the Krylov space (Nazareth, 1979). Our argument requires neither an exact line search nor the BFGS update. Let $\mathcal{S}_t := \operatorname{span}(\{\delta_0, \dots, \delta_t\} \cup \{g_0, \dots, g_t\})$. We claim that the inverse Hessian approximation satisfies $H_t = \gamma_t I + U_t$ with a scalar γ_t and $\operatorname{range}(U_t) \cup \operatorname{range}(U_t^\top) \subseteq \mathcal{S}_t$ for every t. For $t = 0$ we have $U_0 = 0$. Assume that the claim holds for t. Then the search direction $p_t = -H_t g_t = -\gamma_t g_t - U_t g_t$ lies in $\mathcal{S}_t$. A line search multiplies p_t by a scalar α_t, so $\delta_{t+1} = \delta_t + \alpha_t p_t \in \mathcal{S}_t$, which is the linear-span assumption at step t. In a self-scaled variant, H_t is first multiplied by a scalar τ_t, which replaces γ_t and U_t by $\tau_t\gamma_t$ and $\tau_t U_t$, respectively, and so preserves the claim for t. Without loss of generality we may therefore assume that the update is applied to H_t. The differences $s_t := \delta_{t+1} - \delta_t$ and $y_t := g_{t+1} - g_t$ lie in $\mathcal{S}_{t+1}$. Moreover, $H_t y_t = \gamma_t y_t + U_t y_t$ and $H_t^\top y_t = \gamma_t y_t + U_t^\top y_t$ lie in $\mathcal{S}_{t+1}$, since $\mathcal{S}_t \subseteq \mathcal{S}_{t+1}$. Recall that every Broyden-class update, namely BFGS, DFP, SR1, and their combinations with real coefficients, forms H_{t+1} by adding to H_t a sum of terms of the forms

$$a\, s_t s_t^\top, \qquad b\,\big(s_t y_t^\top H_t + H_t y_t s_t^\top\big), \qquad c\, H_t y_t y_t^\top H_t,$$

together with, for SR1, terms of the form $(s_t - H_t y_t)(s_t - H_t y_t)^\top$; see Nocedal & Wright (2006, Ch. 6) for the inverse form of these updates. Every such term has range and co-range in $\mathcal{S}_{t+1}$. Hence $H_{t+1} = \gamma_{t+1} I + U_{t+1}$ with $\operatorname{range}(U_{t+1}) \cup \operatorname{range}(U_{t+1}^\top) \subseteq \mathcal{S}_{t+1}$, which completes the induction. A limited-memory variant forms H_{t+1} from $\gamma_{t+1} I$ by these updates with the stored pairs (s_k, y_k), $k \le t$. All of these pairs lie in $\mathcal{S}_{t+1}$, so the argument above applies to each update.

The self-scaled Broyden method used in our experiments belongs to the Broyden class with $H_0 = \gamma I$. The SSBroyden variants of our implementation initialize $H_0 = I$ and apply at each iteration a Broyden-class rank-two update built from (s_k, y_k) together with a scalar self-scaling factor. The variants differ only in how this scalar is computed. Each variant performs a Wolfe line search along $p_k = -H_k g_k$. If $|y_k^\top s_k|$ is numerically degenerate, no update is made and the previous H_k is retained. Each of these operations preserves the claim of the previous paragraph; for the skipped update, this holds because $\mathcal{S}_k \subseteq \mathcal{S}_{k+1}$. Hence this implementation satisfies the linear-span assumption. A non-scalar H_0, for example a diagonal preconditioner, would instead require $H_0 V_C \subseteq V_C$ for the iterates to stay in V_C. This holds for H_0 a polynomial in the weighted Gram $\widehat{\mathbf{G}}$ of Theorem 4.2, but not for a general diagonal matrix.

Finally, we show that the class of methods satisfying the linear-span assumption does not contain Adam-type diagonal preconditioning. The reason is that, for a diagonal matrix D that is not a multiple of the identity, $Dg \notin V_C$ in general, as the following example shows. Recall that Adam-type methods step along an average of the gradients, scaled coordinatewise by $D_t = \operatorname{diag}(v_t)^{-1/2}$, where v_t is an average of the squared coordinates of the gradients. Here, we set the stabilizing constant to zero. Let $n = 1$, $\varphi(x_1) = (1, 2)^\top$, and $r_0(x_1) \neq 0$, so that $V_C = \operatorname{span}\{(1, 2)^\top\}$ and $g_0 \neq 0$.

Then every gradient is proportional to $(1,2)^\top$. Hence $v_0 \propto (1,4)$ and $D_0 \propto \mathrm{diag}(1,\frac{1}{2})$. Thus $\delta_1 \propto D_0(1,2)^\top \propto (1,1)^\top$, which does not lie in V_C. By the induction at the beginning of this proof, every method satisfying the linear-span assumption has $\delta_1 \in V_C$, so the Adam-type method does not satisfy it. This concludes the proof. □

B.2 Proof of Lemma 3.1(ii) and (iii)

Proof. We begin with part (ii). The matrix Φ_C has rank n_d, since its n_d rows, one per distinct batch point, are linearly independent. Recall that $\ker \Phi_C = \mathrm{range}(\Phi_C^\top)^\perp = V_C^\perp$. Hence $\mathcal{A} = \{\delta : \Phi_C\delta = -r_0|_C\}$ is a nonempty affine subspace with direction $V_C^\perp$. The restriction of $\delta \mapsto \Phi_C\delta$ to V_C is injective: if $\Phi_C\delta = 0$ and $\delta \in V_C$, then $\delta \in V_C \cap V_C^\perp = \{0\}$. As $\dim V_C = \mathrm{rk}\,\Phi_C = n_\mathrm{d}$, this restriction is also surjective onto $\mathbb{R}^{n_\mathrm{d}}$. Thus $\mathcal{A}\cap V_C$ consists of a single point, which we denote by δ^mn. Every $\delta \in \mathcal{A}$ decomposes as $\delta = \delta^\mathrm{mn} + z$ with $z \in V_C^\perp$, so that $\|\delta\|^2 = \|\delta^\mathrm{mn}\|^2 + \|z\|^2$. It follows that δ^mn is the unique minimum-norm interpolant. By the standard characterization of the pseudoinverse as the minimum-norm solution of a consistent linear system, we have $\delta^\mathrm{mn} = -\Phi_C^+ r_0|_C$.

Now assume that a method satisfying the linear-span assumption attains $\widehat{L}(\delta_\infty) = 0$, at an iterate or as the limit of its iterates. Since all $\omega_i > 0$, we have $\delta_\infty \in \mathcal{A}$. Moreover, every iterate lies in V_C by part (i), and the subspace V_C is closed, so $\delta_\infty \in \mathcal{A}\cap V_C = \{\delta^\mathrm{mn}\}$, that is, $\delta_\infty = \delta^\mathrm{mn}$. The point δ^mn depends only on Φ_C and $r_0|_C$, so δ_∞ depends neither on the method, nor on its hyperparameters, nor on the weights. For gradient descent started at $\delta_0 = 0$, the identity $\delta_\infty = \delta^\mathrm{mn}$ is the minimum-norm implicit bias of Gunasekar et al. (2018); Soudry et al. (2018). The argument above is the one for gradient descent, except that part (i) extends the confinement to V_C from gradient descent to every method satisfying the linear-span assumption.

Next, we derive the updated residual field of part (ii),

$$r_0(x) + \big(\Phi\delta^\mathrm{mn}\big)(x) = r_0(x) - k_C(x)^\top \mathbf{K}_C^{-1}\, r_0|_C, \tag{7}$$

where k_C and $\mathbf{K}_C$ are indexed by the distinct batch points. Since the rows of Φ_C are independent, we have $\Phi_C^+ = \Phi_C^\top(\Phi_C\Phi_C^\top)^{-1}$, and $\Phi_C\Phi_C^\top = \mathbf{K}_C$. Hence $(\Phi\delta^\mathrm{mn})(x) = -\varphi(x)^\top\Phi_C^\top\mathbf{K}_C^{-1}r_0|_C = -k_C(x)^\top\mathbf{K}_C^{-1}r_0|_C$. The term $k_C(x)^\top\mathbf{K}_C^{-1}r_0|_C$ is the interpolant of r_0 on C with respect to the empirical tangent kernel (Jacot et al., 2018), here applied to the residual rather than to the network output. The updated residual field is therefore the error of this interpolant.

We turn to part (iii). Here, $W \succ 0$ denotes the $n_\mathrm{d} \times n_\mathrm{d}$ diagonal matrix whose entry for a distinct batch point is the sum of the weights of its copies in the batch. It replaces the $n \times n$ matrix $\mathrm{diag}(\omega_i)$ of the statement, and both give the same objective $\widehat{L}$, since Φ_C carries one row per distinct point. Then $\widehat{L}(\delta) = \frac{1}{n}\left\|W^{1/2}(r_0|_C + \Phi_C\delta)\right\|^2$, and its gradient is $\frac{2}{n}\Phi_C^\top W(r_0|_C + \Phi_C\delta)$. Hence the minimizers of the convex quadratic $\widehat{L}$ are the solutions of the weighted normal equations $\Phi_C^\top W\Phi_C\,\delta = -\Phi_C^\top W r_0|_C$. By the standard characterization of the pseudoinverse, $\delta^\mathrm{ls} = -(W^{1/2}\Phi_C)^+ W^{1/2} r_0|_C$ is the minimum-norm solution of these equations. It lies in $\mathrm{range}((W^{1/2}\Phi_C)^\top) = \mathrm{range}(\Phi_C^\top) = V_C$. As $W \succ 0$, we have $\ker(\Phi_C^\top W\Phi_C) = \ker\Phi_C = V_C^\perp$, so the solutions form the affine set $\delta^\mathrm{ls} + V_C^\perp$. This set meets V_C only in δ^ls: if $\delta^\mathrm{ls} + z \in V_C$ with $z \in V_C^\perp$, then $z \in V_C \cap V_C^\perp = \{0\}$. A method satisfying the linear-span assumption has every iterate in V_C by part (i), and V_C is closed, so a limit of the iterates that minimizes $\widehat{L}$ lies in $(\delta^\mathrm{ls} + V_C^\perp) \cap V_C = \{\delta^\mathrm{ls}\}$. When the system $\Phi_C\delta = -r_0|_C$ is consistent, the solution set of the weighted normal equations equals $\mathcal{A}$ and does not depend on W, which recovers part (ii). This concludes the proof. □

Note that the independence of the rows in part (ii) is sufficient, but not necessary, for $\mathcal{A} \cap V_C$ to be a single point. Indeed, if the system $\Phi_C\delta = -r_0|_C$ is consistent, the last step of the proof of part (iii) shows that $\mathcal{A} \cap V_C = \{\delta^\mathrm{ls}\}$, and δ^ls is then the unique minimum-norm interpolant. Independence additionally guarantees consistency for every right-hand side and the invertibility of $\mathbf{K}_C$ used in (7).

B.3 Proof of Theorem 4.2

Proof. Let $S_\lambda := (\mathbf{G} + \lambda I)^{1/2}$ and $\psi(x) := S_\lambda^{-1}\varphi(x)$, so that $\|\psi(x)\|^2 = \ell_\lambda(x)$, and let

$$X_i := \frac{1}{n}\,\omega(x_i)\,\psi(x_i)\psi(x_i)^\top, \qquad \Sigma_\lambda := \mathbb{E}\Big[\sum_i X_i\Big],$$

where here and below the expectations of the X_i are over the draws $x_i \overset{\text{iid}}{\sim} q\,\mathrm{d}\rho$. Since $\omega = 1/q$, we have $\mathbb{E}_q[\omega\,\psi\psi^\top] = \mathbb{E}_\rho[\psi\psi^\top]$, and hence $\Sigma_\lambda = S_\lambda^{-1}\mathbf{G}S_\lambda^{-1}$. We now reduce (3) to a bound on a centered sum. As $S_\lambda^{-1}(\widehat{\mathbf{G}} + \lambda I)S_\lambda^{-1} = \sum_i X_i + \lambda S_\lambda^{-2}$ and $S_\lambda^{-1}(\mathbf{G} + \lambda I)S_\lambda^{-1} = \Sigma_\lambda + \lambda S_\lambda^{-2} = I$, the embedding (3) is equivalent to $\|\sum_i X_i - \Sigma_\lambda\| \le \varepsilon$.

Next, we bound each summand. The leverage-overestimate condition $q \ge \beta\ell_\lambda/d_\lambda$ gives $\omega(x)\ell_\lambda(x) \le d_\lambda/\beta =: B_\lambda$. Hence $\|X_i\| = \frac{1}{n}\omega(x_i)\ell_\lambda(x_i) \le B_\lambda/n$ almost surely. Moreover, $\|\mathbb{E}X_i\| = \|\Sigma_\lambda\|/n \le 1/n$, since $\Sigma_\lambda = I - \lambda S_\lambda^{-2} \preceq I$. For positive semidefinite A and B we have $\|A - B\| \le \max\{\|A\|, \|B\|\}$, so $\|X_i - \mathbb{E}X_i\| \le \max\{B_\lambda, 1\}/n \le 2B_\lambda/n$, where the last inequality uses $B_\lambda \ge d_\lambda \ge \frac{1}{2}$, which is shown below.

We now bound the variance. Using $A^2 \preceq \|A\|\,A$ for positive semidefinite A and $\mathbb{E}[(X_i - \mathbb{E}X_i)^2] \preceq \mathbb{E}[X_i^2]$, we obtain

$$\mathbb{E}\big[X_i^2\big] \preceq \frac{B_\lambda}{n}\,\mathbb{E}[X_i] = \frac{B_\lambda}{n^2}\,\Sigma_\lambda, \qquad \text{so} \qquad \sum_i \mathbb{E}\big[(X_i - \mathbb{E}X_i)^2\big] \preceq \mathbf{V} := \frac{B_\lambda}{n}\,\Sigma_\lambda.$$

The intrinsic-dimension matrix Bernstein inequality remains valid when the variance is replaced by any positive semidefinite upper bound of it (Tropp, 2015, Thm. 7.3.1 and the remark following it). We use the upper bound $\mathbf{V}$. Since $\Sigma_\lambda = I - \lambda S_\lambda^{-2} \preceq I$, we have $\|\mathbf{V}\| = B_\lambda\,\|\Sigma_\lambda\|/n \le B_\lambda/n$. The intrinsic dimension of $\mathbf{V}$ is $\operatorname{intdim}(\mathbf{V}) = \operatorname{tr}\Sigma_\lambda/\,\|\Sigma_\lambda\| = d_\lambda/\,\|\Sigma_\lambda\| \le 2d_\lambda$. The last inequality holds because $\|\Sigma_\lambda\| = \lambda_{\max}(\mathbf{G})/(\lambda_{\max}(\mathbf{G}) + \lambda) \ge \frac{1}{2}$ under the condition $\lambda \le \lambda_{\max}(\mathbf{G})$ of Theorem 4.2.

It remains to apply the matrix Bernstein inequality. For $t \ge \|\mathbf{V}\|^{1/2} + 2B_\lambda/(3n)$, it bounds the largest eigenvalue of the centered sum (Tropp, 2015, Thm. 7.3.1):

$$\mathbb{P}\Big[\lambda_{\max}\Big(\sum_i X_i - \Sigma_\lambda\Big) \ge t\Big] \;\le\; 4\operatorname{intdim}(\mathbf{V})\,\exp\Big(\frac{-t^2/2}{\|\mathbf{V}\| + 2B_\lambda t/(3n)}\Big).$$

We apply it to $\pm(X_i - \mathbb{E}X_i)$, which have the same $\mathbf{V}$ and the same B_λ, and use $\operatorname{intdim}(\mathbf{V}) \le 2d_\lambda$. A union bound over the two signs gives

$$\begin{aligned}\mathbb{P}\Big[\Big\|\sum_i X_i - \Sigma_\lambda\Big\| \ge t\Big] &\le 16 d_\lambda\,\exp\Big(\frac{-t^2/2}{\|\mathbf{V}\| + 2B_\lambda t/(3n)}\Big)\\ &\le 16 d_\lambda\,\exp\Big(\frac{-n\,t^2}{2B_\lambda(1 + 2t/3)}\Big).\end{aligned}$$

We set $t = \varepsilon \le 1$. The right-hand side is then at most ζ as soon as $n \ge \frac{2}{\varepsilon^2}\big(1 + \frac{2\varepsilon}{3}\big)B_\lambda \log\frac{16d_\lambda}{\zeta}$. This holds under the condition $n \ge 4\varepsilon^{-2}(d_\lambda/\beta)\log(16d_\lambda/\zeta)$ of Theorem 4.2, since $2(1 + 2\varepsilon/3) \le 10/3 < 4$. Moreover, $d_\lambda = \operatorname{tr}\Sigma_\lambda \ge \|\Sigma_\lambda\| \ge \frac{1}{2}$ and $\zeta < 1$, so $\log(16d_\lambda/\zeta) > \log 8 > 1$. Hence the same condition gives $n \ge 4B_\lambda/\varepsilon^2$, so that $\|\mathbf{V}\|^{1/2} + 2B_\lambda/(3n) \le \varepsilon/2 + \varepsilon^2/6 < \varepsilon$. Thus $t = \varepsilon$ lies above the threshold from which the matrix Bernstein inequality applies. With probability at least $1 - \zeta$, we obtain $\|\sum_i X_i - \Sigma_\lambda\| < \varepsilon$, which implies (3), as shown above.

Analogously, we treat uniform sampling. For $q \equiv 1$, we have $\|X_i\| = \ell_\lambda(x_i)/n \le (\sup_x \ell_\lambda)/n$ almost surely. Hence the bound on each summand, the variance bound, and the matrix Bernstein inequality hold with B_λ replaced by $\sup_x \ell_\lambda$, while the intrinsic dimension of $\mathbf{V}$ is unchanged. Together, they give the condition $n \ge 4\varepsilon^{-2}(\sup_x \ell_\lambda)\log(16d_\lambda/\zeta)$ for (3) to hold with probability at least $1 - \zeta$. This concludes the proof. □

B.4 Proof of Corollary 4.3

Proof. Let $f \in H_K$, and let δ be the minimum-norm representer of f, so that $f = \varphi^\top\delta$ and $\|\delta\|_2 = \|f\|_{H_K}$. Let Π_C be the orthogonal projector onto V_C; by the proof of Proposition 4.4(i)

below, $s_C f = \varphi^\top \Pi_C \delta$, so that $f - s_C f = \Phi v$ with $v := (I - \Pi_C)\delta$. Since every summand $\omega(x_i)\varphi(x_i)\varphi(x_i)^\top$ of $\widehat{\mathbf{G}}$ has its range in V_C, we have $\widehat{\mathbf{G}} v = 0$, and the lower bound in (3) applied to v gives

$$(1-\varepsilon)\big(v^\top \mathbf{G} v + \lambda \|v\|_2^2\big) \;\le\; v^\top \widehat{\mathbf{G}} v + \lambda \|v\|_2^2 \;=\; \lambda \|v\|_2^2, \qquad \text{hence} \qquad v^\top \mathbf{G} v \;\le\; \frac{\varepsilon\lambda}{1-\varepsilon}\, \|v\|_2^2 .$$

Then $\|f - s_C f\|_\rho^2 = \|\Phi v\|_\rho^2 = v^\top \mathbf{G} v \le \frac{\varepsilon\lambda}{1-\varepsilon} \|v\|_2^2 \le \frac{\varepsilon\lambda}{1-\varepsilon} \|\delta\|_2^2$, since $\|(I - \Pi_C)\delta\|_2 \le \|\delta\|_2$. The weights enter only through the embedding event. For a batch $C' \supseteq C$, the vector $v' := (I - \Pi_{C'})\delta$ is orthogonal to $V_{C'} \supseteq V_C$, so $\widehat{\mathbf{G}} v' = 0$ for the Gram of the batch of the theorem, and the same inequality holds with $\|v'\|_2 \le \|\delta\|_2$. This concludes the proof. □

B.5 Proof of Proposition 4.4

Proof. We begin with part (i). Let $f \in H_K$, and let δ be any vector with $f = \varphi^\top \delta$. Since $k_C(x) = \Phi_C \varphi(x)$, $\mathbf{K}_C = \Phi_C \Phi_C^\top$, and $f|_C = \Phi_C \delta$, the interpolant of f on C is

$$(s_C f)(x) = \varphi(x)^\top \Phi_C^\top (\Phi_C \Phi_C^\top)^+ \Phi_C \delta = \varphi(x)^\top \Pi_C \delta,$$

where $\Pi_C := \Phi_C^\top (\Phi_C \Phi_C^\top)^+ \Phi_C$ is the orthogonal projector onto the row space V_C, for every batch: $A^\top (A A^\top)^+ A$ is that projector for any matrix A, with or without repeated or dependent rows, and the pseudoinverse is the inverse when the rows of Φ_C are independent. Together with $\kappa(x) = \|\varphi(x)\|^2$, the same identities show that the power function satisfies

$$P_C(x)^2 = \kappa(x) - k_C(x)^\top \mathbf{K}_C^+ k_C(x) = \kappa(x) - \varphi(x)^\top \Pi_C \varphi(x) = \|(I - \Pi_C)\varphi(x)\|^2 .$$

By the Cauchy–Schwarz inequality,

$$\begin{aligned} \big|f(x) - (s_C f)(x)\big| &= \big|\varphi(x)^\top (I - \Pi_C)\delta\big| = \big|\langle (I - \Pi_C)\varphi(x), (I - \Pi_C)\delta\rangle\big| \\ &\le \|(I - \Pi_C)\varphi(x)\|\, \|\delta\| = P_C(x)\, \|\delta\| . \end{aligned}$$

Taking for δ the minimum-norm representer of f gives $\|\delta\| = \|f\|_{H_K}$, which proves part (i).

We turn to part (ii). Let $C_1 \subseteq C_2$. Then $V_{C_1} \subseteq V_{C_2}$, hence $\Pi_{C_1} \preceq \Pi_{C_2}$, and by the identity for P_C^2 in the proof of part (i),

$$\begin{aligned} P_{C_2}^2 &= \|(I - \Pi_{C_2})\varphi\|^2 = \kappa - \varphi^\top \Pi_{C_2} \varphi \\ &\le \kappa - \varphi^\top \Pi_{C_1} \varphi = \|(I - \Pi_{C_1})\varphi\|^2 = P_{C_1}^2 . \end{aligned}$$

Taking square roots gives $P_{C_2} \le P_{C_1}$. Part (iii) follows from part (ii) applied to $C_{\mathrm{cov}} \subseteq C$ and $C_{\mathrm{lev}} \subseteq C$. This concludes the proof. □

C Experimental details and deferred results

C.1 The six benchmarks

In this section we state the six PDEs used as benchmarks, together with the initial and boundary data, the parameters and the time horizon as implemented, and the error metric of each benchmark. All boundaries but the two walls of Rayleigh–Taylor are periodic, and the periodicity is imposed architecturally, so no other boundary residual enters $\widehat{L}$. The surrounding pipeline is that of Section 5 with the domain importance adaptation method of the second phase replaced by the batch rule under test (Appendix C.3).

C.1.1 Burgers equation

The one-dimensional Burgers equation considered has the form

$$\begin{aligned} u_t + u u_x - \nu u_{xx} = 0, &\qquad x \in [-1, 1],\ t \in [0, 1], \\ u(0, x) = -\sin(\pi x), &\qquad x \in [-1, 1], \\ u(t, -1) = u(t, 1), &\qquad t \in [0, 1], \end{aligned}$$

where $u(t, x)$ is the velocity field and $\nu = 1/(100\pi)$ the kinematic viscosity controlling the diffusive term. Unlike the Dirichlet formulation of Raissi et al. (2019), the boundary is periodic here and is built into the network. The performance of this benchmark is the relative L^2 error of u.

C.1.2 ALLEN–CAHN EQUATION

The one-dimensional Allen–Cahn equation considered is the variant of Kiyani et al. (2025),

$$\begin{aligned} u_t - 10^{-4}u_{xx} + 5u^3 - 5u = 0, \qquad & x \in [-1,1],\ t \in [0,1], \\ u(0,x) = x^2\sin(2\pi x), \qquad & x \in [-1,1], \\ u(t,-1) = u(t,1), \quad u_x(t,-1) = u_x(t,1), \qquad & t \in [0,1]. \end{aligned}$$

Allen–Cahn is a standard benchmark for physics-informed training and is hard because it is stiff, with sharp transitions in space and time under a periodic boundary condition. The performance of this benchmark is the relative L^2 error of u.

This is the only Allen–Cahn problem in this paper. We do not report the more common variant with $u(0,x) = x^2\cos(\pi x)$ (Raissi et al., 2019), for one reason: that initial condition is continuous but not differentiable across the periodic boundary ($u_x(-1) = 2$, $u_x(1) = -2$), so its Fourier coefficients decay as k^{-2} and the spectral reference used throughout the literature (511–512 Fourier modes, ETDRK4, $\Delta t = 10^{-5}$) converges only at second order in the number of modes. Against a 4,096-mode solve the two published reference files carry a relative L^2 error of 1.1×10^{-5}, concentrated in $t < 0.15$, while agreeing with each other to 1.1×10^{-6}. A refinement sequence of our own (511 to 32,704 modes, self-convergence 1.2×10^{-5} to 2.7×10^{-9} against a 65,408-mode solve, order $\approx N^{-2.2}$) confirms the rate. Certifying an error of 10^{-9} on that variant therefore takes $64\times$ the standard resolution, and published numbers below 10^{-5} on it are measured against references less accurate than themselves. The initial condition of Kiyani et al. (2025) is C^1-periodic; with it the reference converges spectrally once the interfaces are resolved (6.8×10^{-9} at 1,024 modes and 4.0×10^{-10} at 2,048, against 4,096).

C.1.3 KORTEWEG–DE VRIES EQUATION

The one-dimensional Korteweg–de Vries equation is expressed as

$$\begin{aligned} u_t + \eta u u_x + \mu^2 u_{xxx} = 0, \qquad & x \in [-1,1],\ t \in [0,1], \\ u(0,x) = \cos(\pi x), & \\ u(t,-1) = u(t,1), & \end{aligned}$$

where u is the wave amplitude, η governs the strength of the nonlinearity and μ the dispersion. Under the KdV dynamics the initial wave evolves into a series of solitary-type waves. Here $\eta = 1$ and $\mu = 0.022$, the Zabusky–Kruskal parameterization, so the dispersive coefficient is $\mu^2 = 4.84 \times 10^{-4}$. The performance of this benchmark is the relative L^2 error of u.

C.1.4 KURAMOTO–SIVASHINSKY EQUATION

The one-dimensional Kuramoto–Sivashinsky equation takes the form

$$\begin{aligned} u_t + \alpha u u_x + \beta u_{xx} + \gamma u_{xxxx} = 0, \qquad & x \in [0,2\pi],\ t \in [0,T], \\ u(0,x) = u_0(x), & \end{aligned}$$

where u is the height of a thin film or flame front. The equation arises in flame-front propagation, thin-film flows and plasma instabilities. Here $u_0(x) = \cos(x)\big(1 + \sin(x)\big)$ and $\alpha = \frac{100}{16}$, $\beta = \frac{100}{16^2}$, $\gamma = \frac{100}{16^4}$. The realized horizon is $T = 0.344$, one stored slice short of the 0.35 defined by Wang et al. (2026). The performance of this benchmark is the relative L^2 error of u.

C.1.5 GINZBURG–LANDAU EQUATION

The two-dimensional complex Ginzburg–Landau equation takes the form

$$\begin{aligned} A_t - \epsilon\Delta A - \mu A + \gamma A|A|^2 = 0, \qquad & (x,y) \in (-1,1)^2,\ t \in [0,T], \\ A_0(x,y) = (10y + 10ix)\exp\big(-25(x^2+y^2)\big), & \end{aligned}$$

with periodic boundary conditions on both spatial axes. Here $A = u + iv$ is the complex amplitude representing the envelope of oscillations, ϵ the (real) diffusion coefficient, μ the linear growth rate and γ the nonlinear saturation. The parameters are $\epsilon = 0.004$, $\mu = 10$ and $\gamma = 10 + 15i$, equivalently

$\mu\big[A-(1+\frac{3}{2}i)|A|^2A\big]$ with $\mu = 10$. The equation is integrated as the two-field real system (u, v), so its residual contributes two rows per collocation point. The realized horizon is $T = 0.79$, one stored slice short of the requested 0.8. As on Kuramoto–Sivashinsky the network receives the rescaled time t/T. The performance of this benchmark is the relative L^2 error of the real part u.

C.1.6 Rayleigh–Taylor instability

The coupled flow–temperature system modeling buoyancy-driven instabilities takes the form

$$\begin{aligned} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} &= -\nabla p + \sqrt{\tfrac{\mathrm{Pr}}{\mathrm{Ra}}}\,\Delta\boldsymbol{u} + T\boldsymbol{e}_y, \\ \nabla\cdot\boldsymbol{u} &= 0, \\ T_t + (\boldsymbol{u}\cdot\nabla)T &= \tfrac{1}{\sqrt{\mathrm{Pr}\,\mathrm{Ra}}}\,\Delta T, \end{aligned}$$

with the velocity $\boldsymbol{u} = (u, v)$ on $(x, y) \in [0, 1] \times [0, 2]$. Here T is the temperature field, Pr the Prandtl number and Ra the Rayleigh number. The system produces the characteristic mushroom-shaped plumes as the heavier fluid penetrates the lighter one below it. The boundary conditions are periodic in the horizontal direction for u, v and T, with $u = v = T = 0$ on the walls $y = 0$ and $y = 2$. The periodicity is built into the network, and the wall conditions enter the loss as a boundary term. The parameters are $\mathrm{Pr} = 0.71$ and $\mathrm{Ra} = 10^6$. Unlike the five benchmarks above, which are solved from $t = 0$, Rayleigh–Taylor is solved on the window $t \in [1.01, 5.76]$ of its reference trajectory, shifted to start at 0: the initial condition is the reference solution at $t = 1.01$, and the horizon is $T = 4.75$. The performance of this benchmark is the relative L^2 error of the temperature field T. The error logged by the runs is the mean over the three fields and is not the quantity reported here.

Realized time horizons. For Ginzburg–Landau, KdV and Kuramoto–Sivashinsky we take the reference data, the loading routine and the time ranges of Wang et al. (2026) unchanged. That loader retains the stored time slices lying strictly below the requested bound, so the horizon actually trained and evaluated on is one slice short of the nominal one: the requested 0.8 on Ginzburg–Landau retains 80 of 101 slices and realizes $t \in [0, 0.79]$, and the requested 0.35 on Kuramoto–Sivashinsky retains 87 of 251 slices and realizes $t \in [0, 0.344]$. The horizons stated above are the realized ones, and all errors in this paper are computed on exactly those grids. The distinction is immaterial to any claim we make, but it matters when our numbers are set beside published ones on the same equations, since a shorter horizon is an easier problem.

C.2 Reference solutions

Every reported error is relative to a reference solution whose own accuracy bounds what the error can resolve. Table 4 states, per benchmark, the generator, the resolution and the self-convergence of the reference, that is, its relative L^2 distance to a solve at $2\times$ the resolution and half the time step (Burgers: $4{,}088$ modes and $\Delta t = 1.25 \times 10^{-6}$, and Allen–Cahn: $16{,}384$ modes). An independent NumPy ETDRK4 verifier, released with the code, reproduces the shipped Burgers and Allen–Cahn files at matched resolution to 10^{-14}–10^{-11}.

The margins matter where the reported errors are smallest: the Burgers reference is certified at 1.9×10^{-11}, about $17\times$ below the best median of Table 1, so those numbers measure the network and not the reference; on Allen–Cahn the margin is three orders (1.1×10^{-7} against 5.9×10^{-11}), on Kuramoto–Sivashinsky $\approx 70\times$; on Ginzburg–Landau $\approx 1{,}600\times$, and on KdV and Rayleigh–Taylor about five orders. The Burgers reference is the refined file, not the 511-mode file of the public benchmark set (relative error 4×10^{-5}). The Rayleigh–Taylor reference is a Fourier–Chebyshev solve of the vorticity–streamfunction form; a second, independent solver in primitive variables on the same grid and time step agrees with it to 6.3×10^{-9} in T over the window, the entry of its last column.

C.3 Shared pipeline configuration

Every method runs the same pipeline; only the rule that draws the frozen batch of the second phase changes. Tables 5 and 6 give the per-benchmark settings, and the paragraphs after them the parts that are common to all six or do not tabulate. The fields are taken from the run configurations and the code.

Table 4: Reference solutions: generator, resolution, precision and self-convergence (relative L^2 against a finer solve. Rayleigh–Taylor: against an independent second solver at the same resolution). The reported errors of Table 1 are resolved by the reference only above the last column. The Burgers grid ($2{,}044 = 4 \times 511$ modes) contains the 512-point evaluation grid exactly, so no interpolation enters its error.

benchmark	solver	resolution	Δt	prec.	self-convergence
Burgers	Fourier ETDRK4 (Kassam–Trefethen), NumPy	2,044 modes	2.5×10^{-6}	f64	1.9×10^{-11}
Allen–Cahn (sine)	Fourier ETDRK4, NumPy	8,176 modes	10^{-5}	f64	5.9×10^{-11}
Ginzburg–Landau	Chebfun `spin2`	200×200 on $[-1, 1]^2$	10^{-3}	f64	3.2×10^{-8}
KdV	Chebfun `spin`	511 modes on $[-1, 1]$	10^{-5}	f64	5.5×10^{-12}
Kuramoto–Sivashinsky	Chebfun `spin`	511 modes on $[0, 2\pi]$	10^{-5}	f64	3.0×10^{-9}
Rayleigh–Taylor	Fourier–Chebyshev IMEX SBDF3	384×512 on $[0, 1] \times [0, 2]$	1.5625×10^{-4}	f64	6.3×10^{-9}

Table 5: Neural network hyperparameters for the benchmark PDEs. The architecture is one multilayer perceptron (MLP) of five layers of width 64 throughout, with a random Fourier feature embedding of dimension 64 whose scale is the initial standard deviation of a trainable kernel. Periodicity is imposed by the hard embedding $[\cos \omega x, \sin \omega x]$ on the axes listed, so no boundary term enters the loss except on the two walls of Rayleigh–Taylor. Physics-informed initialization is applied to the last layer everywhere but on Kuramoto–Sivashinsky and Rayleigh–Taylor, which therefore keep their last-layer bias. Parameter counts are taken from the run logs. N is the number of rows of the second-phase least-squares problem: the residual rows of the batch (two per point on Ginzburg–Landau, four on Rayleigh–Taylor), the initial-condition rows of the full reference grid and, on Rayleigh–Taylor, the wall rows.

	Burgers	**AC**	**GL**	**KdV**	**KS**	**RT**
Architecture						
Activation	Tanh	Tanh	Swish	Tanh	Tanh	Swish
Fourier features scale	1.0	1.0	1.0	1.0	2.0	1.0
Depth $\times$ width	5×64	5×64	5×64	5×64	5×64	5×64
Outputs	1	1	2	1	1	4
Parameters P	20,960	20,960	21,088	20,960	20,961	21,188
Rows N at $n = 16{,}384$	16,896	16,896	112,768	16,896	16,896	223,840
Physics-informed init	yes	yes	yes	yes	no	no
Hard constraint						
Axis	x	x	(x, y)	x	x	x
Periodicity ω	π	π	(π, π)	π	1	2π
Rescaled time input	no	no	yes	no	yes	yes

Table 6: Optimizer and training hyperparameters for the benchmark PDEs. The first phase is SOAP on a fresh uniform batch per iteration; the second is self-scaled Broyden on one frozen batch. The learning-rate schedule is a linear warm-up followed by exponential decay, so on the four 2,000-iteration benchmarks the whole first phase is warm-up and the decay never engages. The initial-condition family is not sampled: its loss is the deterministic mean over the full reference grid at every iteration of both phases, carrying its GradNorm weight.

	Burgers	**AC**	**GL**	**KdV**	**KS**	**RT**
Learning rate schedule						
Initial learning rate	3×10^{-3}					
Decay rate	9×10^{-1}					
Decay steps	2×10^{3}					
Warm-up steps	2×10^{3}					
First-order phase (SOAP)						
Iterations T_1	2k	2k	20k	2k	2k	20k
Residual points per iteration	8,192, drawn i.i.d. uniform, fresh each iteration					
Second-order phase (SSBroyden2)						
Iterations T_b	40,000					
Frozen batch n	16,384					
Candidate pool $M = 4n$	65,536					
Batch refreshed	never (the refresh baselines of Table 7 excepted)					
Loss						
Initial-condition grid	512	512	200×200	512	512	100×200
Boundary term	none (periodicity is architectural)					walls $y \in \{0, 2\}$
Initial weight, residual	1					
Initial weight, initial condition	1	1	100 (both fields)	10	1,000	100 (u, v), 10 (T)

Loss balancing. GradNorm-style equation weights (Wang et al., 2023), momentum 0.9, recomputed at every 1,000th iteration of the first phase before its last (on the four 2,000-iteration benchmarks this fires exactly once, at iteration 1,000, and 19 times on the 20,000-iteration phases) and frozen during the second phase.

The second phase. Self-scaled Broyden (Urbán et al., 2025; Kiyani et al., 2025), variant SSBroyden2 (the three variants differ only in the scalar self-scaling factor), dense $H_0 = I$ ($P \times P$, 3.5 GB in float64), strong-Wolfe line search ($c_1 = 10^{-4}$, $c_2 = 0.9$, at most 25 trial steps) with two fallbacks ($c_2 = 0.8$, then 0.5). The budget is $T_b = 40{,}000$ iterations.

Residual families. The initial-condition family is not sampled but evaluated deterministically on the full reference grid at every iteration of both phases, carrying its GradNorm weight. There is no boundary family except the wall term of Rayleigh–Taylor, since periodicity is built into the network (Remark A.3). The wall rows of Table 5 are evaluated on the wall grid of the reference at every iteration of both phases, as the initial-condition rows are.

Precision and hardware. Float64 in both phases (64-bit JAX enabled before any array is built, highest matmul precision, deterministic cuDNN kernels). One NVIDIA A100-SXM4-80GB per run on a shared cluster. Every runtime quoted in this paper is measured on the A100 (Appendix C.6).

Seeds and statistics. Seeds are first ten primes on all six benchmarks. The batch sizes run in Appendix C.4 use six seeds per size (Table 8). The warmup checkpoint of a seed is shared by every method of that seed. Cells are medians over the seeds a method ran, with the seed spread $\max / \min$. Paired statistics are the per-seed ratio on the shared warmup checkpoints, their median, and the 95 % percentile bootstrap of the median log-ratio. Bold marks the lowest median of a column and every method within $1.3\times$ of it. * marks a method below the uniform default on every paired seed. The paired ratios of SLED to four of the frozen methods are listed in Appendix C.7. The counts of seeds below one in Table 11 use the unrounded per-seed ratios.

Distinct points and the uniform mixture component. LEV and SLED draw their leverage points with replacement, so the number of distinct points n_{d} of a batch lies below n. At $n = 16{,}384$, the seed medians of n_{d} are 13,703.5 / 9,790.5 / 8,873 / 12,990 / 10,309 / 12,078 for LEV and 13,322 / 12,865 / 12,971 / 16,310 / 16,293 / 16,381 for SLED on Allen–Cahn, Ginzburg–Landau, Burgers, KdV, Kuramoto–Sivashinsky and Rayleigh–Taylor, respectively. These medians are 54–84 % and 78.52–99.98 % of the batch, respectively. LEV is not SLED at $\mu_{\mathrm{sob}} = 0$: LEV draws with the uniform mixture component of 0.2 of Table 7, whereas SLED at $\mu_{\mathrm{sob}} = 0$ draws leverage points alone, without a uniform mixture component. At $n = 16{,}384$ the pool size $M = 4n$ of Algorithm 1 is the 65,536 of Tables 6 and 7.

PINNACLE-K. PINNACLE-K runs the selection code of Lau et al. (2024) once, on the warmup checkpoint. The pool has $4n$ fresh uniform points, against 800 PDE points and 200 per initial or boundary family in all but two of their problems. The reference set is the first 1,024 pool points and the eigenvalue cutoff is relative, $10^{-10}\nu_{\max}$. Their runs weight every loss term by one, whereas the kernel of PINNACLE-K scales each residual row by the square root of its GradNorm weight after the warmup (Table 7) similar to how other domain importance adaption methods in our training pipeline behave.

Table 7: The batch rules. A scored rule evaluates its score s on one pool of $M = 65{,}536$ uniform points and draws n points from it with replacement, with probability $q_j = (1 - \lambda_{\mathrm{fl}})\, s_j / \sum_k s_k + \lambda_{\mathrm{fl}}/M$, a uniform mixture component of weight $\lambda_{\mathrm{fl}} = 0.2$, and weight $\omega_j = 1/(Mq_j)$. RAD, SLED and PINNACLE-K depart from this as noted. The settings are the same on every benchmark.

rule	score on the pool	notes
UNIF	—	n i.i.d. uniform points drawn once, no pool; unweighted; the default
SOBOL	—	scrambled Sobol set of n points, no pool; unweighted
LEV	$\widehat{\ell}_\lambda$, Nyström ridge leverage of the tangent features	$L = 1{,}024$ landmarks, $k = 256$ eigenpairs, ridge $\lambda = 10^{-4}\lambda_{\max}$
SLED (Alg. 1)	$\widehat{\ell}_\lambda$ as for LEV, without the uniform mixture component	$\mathrm{round}(\mu_{\mathrm{sob}} n)$ scrambled Sobol points at weight 1, $\mu_{\mathrm{sob}} = \widehat{\rho}$; the rest drawn with $q_j \propto \widehat{\ell}_\lambda(z_j)$ and $\omega_j = 1/(Mq_j)$
RAD k	$\lvert r_0\rvert^k / \mathrm{mean}\lvert r_0\rvert^k + c$	$k = 1$, $c = 1$ (Wu et al., 2023); c is itself a uniform mixture component of weight $c/(1+c) = 0.5$; unweighted
vRBA	—	n pool points drawn uniformly with replacement, weighted by the residual-based attention of Toscano et al. (2026b): $\omega = a^2$, $a = \gamma\,\lvert r_0\rvert / \max\lvert r_0\rvert$, $\gamma \le 20$ annealed
KD-W	$\widehat{\kappa}$, 8 Hutchinson probes	Section 4.2
PINNACLE-K	alignment embedding of Lau et al. (2024): the residual applied to Nyström eigenpairs of the tangent kernel on a reference set	$x \mapsto (\nu_i^{-1/2} v_i^\top K(Z_{\mathrm{ref}}, x)\, r(x))_i$ over every eigenpair (ν_i, v_i) with $\nu_i > 10^{-10}\nu_{\max}$ (283–3,461 kept); its own pool of $4n$ fresh uniform points; Z_{ref} the first 1,024 pool points; the residual rows of all PDE equations, each scaled by the square root of its loss weight; greedy K-means++ seeding of n distinct points in the embedding, once after the warmup; no mixture component; unit weights
refresh	—	batch redrawn every T_r iterations: 500 for RANDOM-R (uniform draws) and RAD, 100 for vRBA, whose anneal advances at each redraw; quasi-Newton memory kept

C.4 BATCH-SIZE SCALING ON THE SIX BENCHMARKS

In this section we provide the medians of the experiments behind Section 5.3.

Table 8: Final relative L^2 error against the batch size n: the medians over seeds drawn as points in Figure 1, in the setup of Section 5.3. n_{Lev} is the sample-complexity bound of Theorem 4.2 for a leverage batch. $^{\dagger k}$ marks k seeds above the accuracy cutoff of Figure 1. – marks a size not run. Underlined: the point of diminishing returns of a method (Section 5.1), the smallest batch size run after which every later doubling improves its median error by less than $2\times$. A row without an underline still improves by at least $2\times$ over its last doubling.

	Allen–Cahn, n_{Lev}=5.6k				Ginzburg–Landau, n_{Lev}=7.6k				Kuramoto–Siv., n_{Lev}=17k					
n	2k	4k	8k	16k	2k	4k	8k	16k	2k	4k	8k	16k	32k	64k
UNIF	2.9e-4	2.5e-5	3.6e-6	2.1e-6	7.0e-2	3.1e-2	1.1e-2	3.3e-3	1.3e-1$^{\dagger 6}$	1.3e-1$^{\dagger 6}$	8.1e-2$^{\dagger 6}$	1.2e-6	5.9e-7	–
SOBOL	1.1e-4	9.1e-6	2.4e-6	4.3e-7	5.3e-2	2.9e-2	7.0e-3	1.4e-3	1.4e-1$^{\dagger 6}$	9.1e-2$^{\dagger 6}$	3.0e-6	2.2e-7	1.3e-7	–
RAD k=1	1.9e-4	1.9e-5	1.0e-5	5.9e-7	2.0e-2	4.6e-3	1.2e-3	2.1e-4	1.3e-1$^{\dagger 6}$	1.2e-1$^{\dagger 6}$	8.9e-2$^{\dagger 6}$	7.8e-6$^{\dagger 1}$	1.5e-6	–
LEV	1.3e-5	4.7e-6	5.6e-7	1.6e-7	1.4e-2	6.9e-4	1.0e-4	5.5e-5	1.3e-1$^{\dagger 6}$	1.2e-1$^{\dagger 6}$	1.0e-1$^{\dagger 6}$	2.7e-3$^{\dagger 5}$	1.2e-6	–
SLED	1.3e-5	2.4e-6	1.4e-7	1.1e-7	4.0e-3	3.2e-4	6.2e-5	4.1e-5	1.2e-1$^{\dagger 6}$	6.9e-2$^{\dagger 5}$	1.0e-6	3.7e-7	3.8e-7	–
	Burgers, n_{Lev}=4.4k				KdV, n_{Lev}=11k				Rayleigh–Taylor, n_{Lev}=27k					
n	2k	4k	8k	16k	2k	4k	8k	16k	2k	4k	8k	16k	32k	64k
UNIF	3.2e-3$^{\dagger 1}$	5.5e-5	7.9e-6	1.7e-7	1.8e-3	9.5e-4	9.8e-7	6.8e-7	2.3e-1$^{\dagger 5}$	2.1e-1$^{\dagger 5}$	1.7e-1$^{\dagger 5}$	5.9e-4	6.1e-4$^{\dagger 1}$	7.0e-4
SOBOL	1.9e-3	5.0e-5	1.2e-6	2.3e-8	1.0e-3	5.1e-4	5.7e-7	6.7e-7	2.0e-1$^{\dagger 5}$	1.7e-1$^{\dagger 5}$	1.2e-1$^{\dagger 4}$	4.9e-4	6.1e-4	7.5e-4
RAD k=1	6.5e-6	9.7e-8	1.0e-8	6.3e-10	6.8e-4	1.6e-3	1.1e-6	8.6e-7	2.2e-1$^{\dagger 5}$	2.4e-1$^{\dagger 5}$	1.6e-1$^{\dagger 4}$	7.3e-4$^{\dagger 2}$	7.0e-4	1.0e-3
LEV	3.2e-7	8.6e-9	2.7e-9	5.6e-10	4.1e-3	3.2e-3	9.3e-7	7.0e-7	2.8e-1$^{\dagger 5}$	4.0e-1$^{\dagger 5}$	4.7e-1$^{\dagger 5}$	2.8e-1$^{\dagger 6}$	1.1e-3$^{\dagger 2}$	8.1e-4
SLED	1.4e-7	5.9e-9	1.0e-9	5.2e-10	7.6e-4	4.9e-6	7.0e-7	6.7e-7	2.0e-1$^{\dagger 5}$	1.9e-1$^{\dagger 5}$	6.2e-2$^{\dagger 3}$	5.0e-4$^{\dagger 1}$	5.4e-4	7.0e-4

Rosters and cutoffs. Table 8 is the build of the batch sizes run, on the seeds first six primes, with failures counted at the accuracy cutoff of Figure 1. Table 1 is the ten-seed roster of Appendix C.3. Therefore, the 16k row in Table 8 slightly differs from the results in Table 1. The point of diminishing returns of a method (Section 5.1) is determined from the rows of Table 8 and is underlined there.

Points to a fixed accuracy. Read as points to a fixed accuracy within the six-seed roster of Table 8, SLED reaches the error of the frozen default at 16,384 with $2\times$ fewer points on Allen–Cahn, $4\times$ on Ginzburg–Landau, $8\times$ on Burgers and $2\times$ on Kuramoto–Sivashinsky ($4\times$ and $1\times$ against the ten-seed medians of Table 1).

The two sample-complexity bounds without constants. Without the theorem's constants, the two sample-complexity bounds of Theorem 4.2 are $\widehat{d_\lambda}\log\widehat{d_\lambda}$ for a leverage batch and $\sup_S \widehat{\ell_\lambda}\log\widehat{d_\lambda}$ for a uniform one, estimated as in Remark A.8. The logarithm is common to both, so their ratio is the coherence $\sup_S \widehat{\ell_\lambda}/\widehat{d_\lambda}$ of Table 1, 61–134 on Kuramoto–Sivashinsky and 23–32 on Rayleigh–Taylor.

Paired statistics above $n = 16{,}384$. On Kuramoto–Sivashinsky, LEV improves by 1,994$\times$ in the paired median for the doubling from 16,384 to 32,768 points, and on Rayleigh–Taylor by 153$\times$ for the same doubling. Above 16,384, on Kuramoto–Sivashinsky and Rayleigh–Taylor, SLED lies below LEV on 6/6 seeds at every batch size run, with paired medians of 0.54$\times$ at 32,768 and 0.80$\times$ at 65,536 points on Rayleigh–Taylor. At $n = 32{,}768$ on Kuramoto–Sivashinsky, the paired median error ratio of SLED to the Sobol set is 3.21$\times$ [2.17, 6.14].

C.5 EXTERNAL REFERENCE POINTS

Table 9 places the errors of Table 1 beside published errors on the same equations, for orientation only: the pipelines are not matched, and no ranking is drawn from it. GL, KdV, KS and RT follow the configurations of Wang et al. (2026), and the sine variant of AC that of Kiyani et al. (2025). Each fixes the equation, the initial and boundary data and the time window. Burgers uses the common equation and initial condition with a periodic instead of a Dirichlet boundary (Appendix C.1). We solve GL on $t \in [0, 0.79]$ and RT on the window $t \in [1.01, 5.76]$ of the reference, shifted to start at 0, both as in the public `jaxpi2` configurations. The errors of Wang et al. (2026) are from their Table 1 (arXiv v1), rounded to two digits.

Table 9: Published final relative L^2 errors on the same equations beside the median error of SLED in Table 1, for orientation only (the pipelines are not matched). Not listed: AC with the $x^2\cos(\pi x)$ initial condition (Appendix C.1.2), and the KdV results of Urbán et al. (2025) and Toscano et al. (2026a), which solve the two-soliton problem with Dirichlet and Neumann boundaries.

Benchmark	SLED	Published	Source
Burgers	3.2×10^{-10}	1.5×10^{-8}	Toscano et al. (2026b)
		1.8×10^{-9}	Toscano et al. (2026a)
Allen–Cahn	1.1×10^{-7}	9.4×10^{-7}	Kiyani et al. (2025)
Ginzburg–Landau	5.1×10^{-5}	7.8×10^{-3}	Wang et al. (2026)
KdV	5.8×10^{-7}	2.3×10^{-4}	Wang et al. (2026)
Kuramoto–Siv.	3.7×10^{-7}	1.1×10^{-2}	Wang et al. (2026)
Rayleigh–Taylor	5.0×10^{-4}	3.8×10^{-3}	Wang et al. (2026)

Table 10: Wall-clock time in seconds on one NVIDIA A100-SXM4-80GB, median over the runs of each cell (2–10). The paired rows divide the second phase of SLED by that of the protocol on the same seed: median and 95 % bootstrap interval.

	Allen–Cahn	Ginzburg–Landau	Burgers	KdV	Kuramoto–Siv.	Rayleigh–T.
batch construction, LEV	8	12	6	11	14	12
batch construction, SLED (probe / total)	73 / 83	59 / 77	27 / 34	83 / 99	105 / 128	65 / 89
second phase, frozen batch (SLED)	1971	2807	1669	2449	3139	3453
second phase, RANDOM-R, T_r=500	1943	2842	1681	2394	3058	3465
second phase, RAD k=1, T_r=500	1963	2831	1657	2382	3097	3488
build / second phase, SLED (%)	4.2	2.8	2.1	4.1	4.1	2.6
SLED / RANDOM-R-500, paired [95 %]	1.03 [1.01, 1.03]	1.00 [0.96, 1.04]	1.00 [0.99, 1.01]	1.03 [1.01, 1.10]	1.02 [1.00, 1.10]	1.00 [0.99, 1.03]
SLED / RAD-500, paired	1.01 [0.99, 1.02]	1.00 [0.97, 1.05]	1.01 [1.00, 1.02]	1.03 [1.02, 1.10]	1.02 [1.01, 1.08]	0.99 [0.98, 1.02]

C.6 COST

Table 10 supports two statements about the batch and nothing more. First, the batch is built once, and its build takes 2.1–4.2 % of the second phase, as a range of the per-benchmark medians over the six benchmarks. The build of LEV takes 6–14 s and that of SLED 34–128 s. Most of the difference is the Hutchinson probe. Second, in the paired median the second phase of SLED lies within 4 % of those of the two refresh protocols it was timed against, RANDOM-R-500 and RAD-500, at paired ratios 0.99–1.03 over 12 cells. Of the 12 paired intervals, 5 lie entirely above 1, with paired medians 1 to 3 % above it, and none lies entirely below 1. The table therefore supports agreement to within 4 % in the paired median and not equality: where the difference is resolved, the second phase of SLED is the slower one, by 1 to 3 %. The runtimes are seconds on the A100.

On Rayleigh–Taylor, whose second phase is the longest of the six, the build takes 89 s of a second phase of 3453 s, a share of 2.6 % within the range of the other five. The paired ratios of this second phase against RANDOM-R-500 and RAD-500 are 1.00 [0.99, 1.03] and 0.99 [0.98, 1.02], respectively, and both intervals contain 1.

C.7 PAIRED RATIOS AT $n = 16{,}384$

Table 11 collects the paired ratios of SLED to UNIF, frozen RAD, SOBOL and LEV, which Section 5.4 quotes. The failed seed of SLED on Rayleigh–Taylor, one of 10 (error 0.17), reproduces on a re-run of the same frozen batch.

C.8 THE BATCHES AFTER THE WARMUP

Figures 2–5 draw the batch of every frozen method of Table 1 on the four benchmarks with one space dimension, for seed 2 at $n = 16{,}384$, each built on the warmup checkpoint of that seed. vRBA is a refresh protocol and its panel shows the first of its batches, which the protocol redraws every 100 iterations (Table 7). The colour of a point is its weight ω_i in the objective (1), with repeated draws of a point summed as in the matrix W of Lemma 3.1. UNIF, SOBOL, RAD and PINNACLE-K are unweighted.

Table 11: Paired ratios of the final error of SLED to that of four other frozen methods at $n = 16{,}384$: median over the shared seeds of the per-seed ratio, the $95\,\%$ percentile bootstrap interval in brackets, and the number of seeds on which the ratio lies below one.

SLED against	Allen–Cahn	Ginzburg–Landau	Burgers	KdV	Kuramoto–Siv.	Rayleigh–T.
UNIF (frozen default)	0.21 [0.05, 0.49]	0.02 [0.01, 0.03]	0.00 [0.00, 0.01]	0.90 [0.77, 1.02]	0.37 [0.24, 0.66]	0.95 [0.74, 1.10]
seeds below one	9/10	10/10	10/10	8/10	10/10	5/10
frozen RAD	0.12 [0.03, 0.54]	0.21 [0.09, 0.58]	0.30 [0.12, 0.57]	0.70 [0.59, 0.88]	0.05 [0.00, 0.19]	0.67 [0.00, 0.89]
seeds below one	8/10	8/10	10/10	10/10	10/10	8/10
SOBOL	0.16 [0.13, 0.40]	0.03 [0.03, 0.05]	0.01 [0.01, 0.03]	0.96 [0.80, 1.10]	1.37 [0.91, 2.60]	1.18 [0.99, 1.65]
seeds below one	10/10	10/10	10/10	5/10	3/10	2/10
LEV	0.77 [0.51, 1.27]	0.94 [0.85, 1.23]	0.75 [0.48, 1.10]	0.85 [0.64, 1.03]	0.00 [0.00, 0.10]	0.00 [0.00, 0.22]
seeds below one	7/10	7/10	7/10	7/10	10/10	10/10

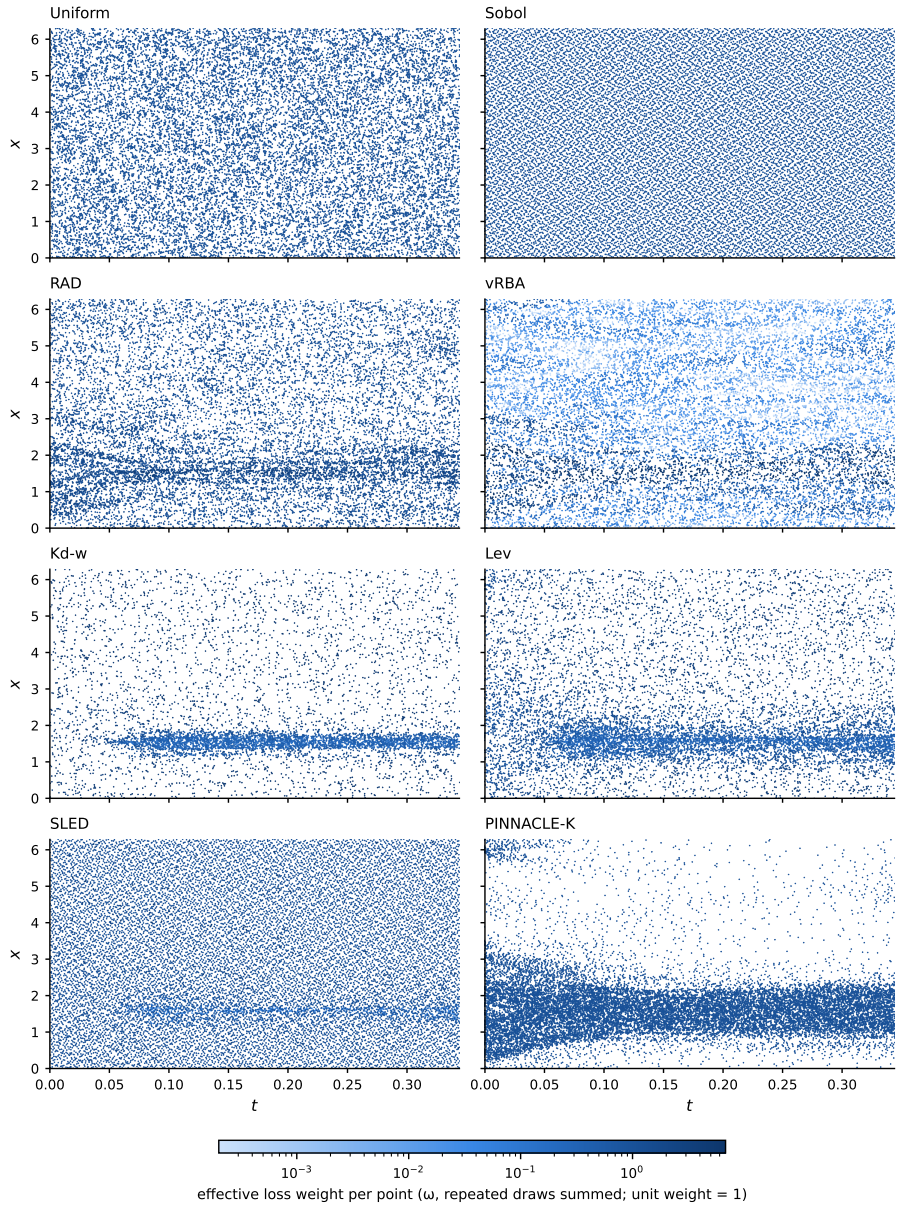

Figure 2: The batches after the warmup on Kuramoto–Sivashinsky: the seven frozen methods of Table 1 and the first batch of vRBA, seed 2, $n = 16{,}384$, built on the warmup checkpoint of that seed. Colour: the weight ω_i of a point in the objective (1), repeated draws of a point summed, logarithmic and scaled per figure. KD-W, LEV and PINNACLE-K concentrate their points in a band around $x \approx 1.5$. SLED places its leverage points in that band and gives $\widehat{\rho} = 0.90$ of its batch to the Sobol component.

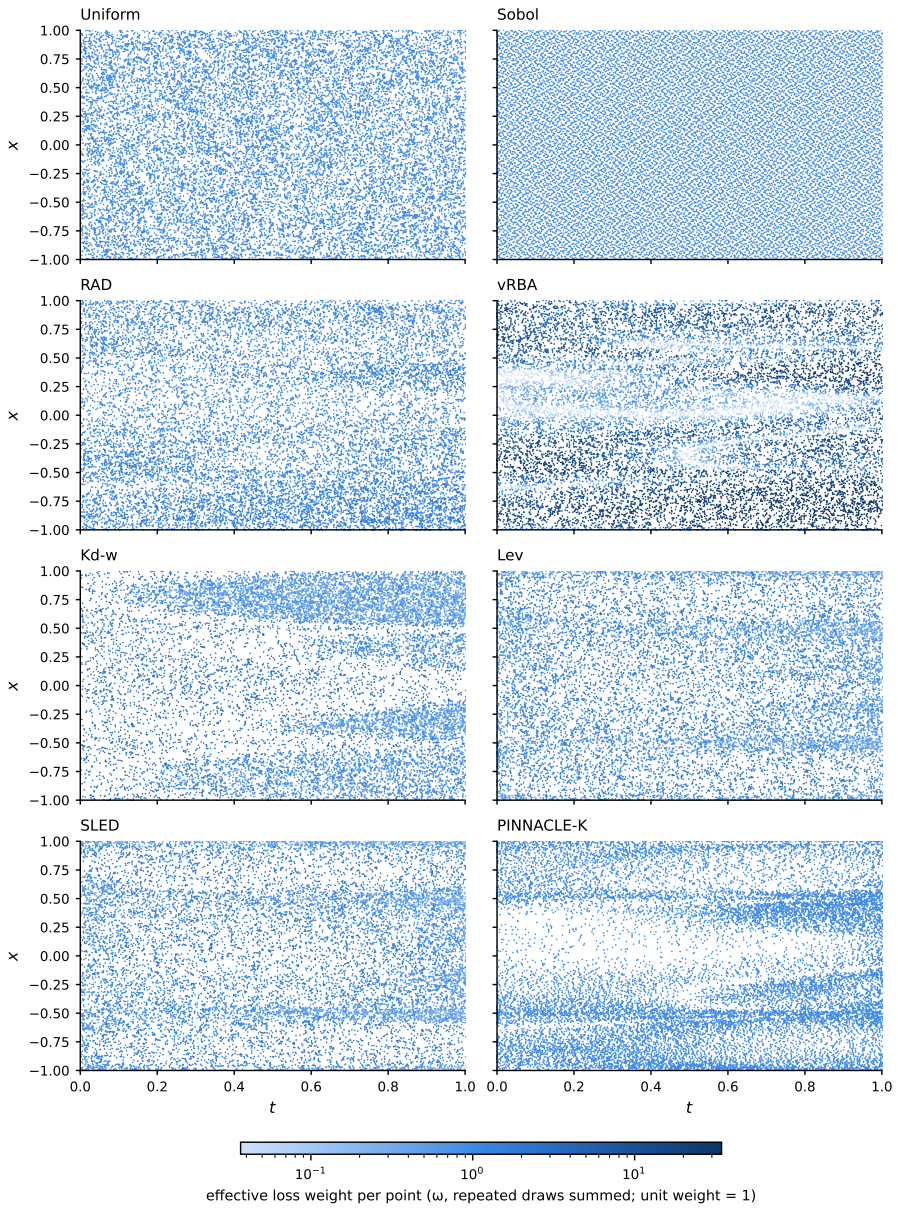

Figure 3: The batches after the warmup on Allen–Cahn, as in Figure 2. SLED has no Sobol component on this seed ($\widehat{\rho} = 0$, the raw rank correlation being -0.19).

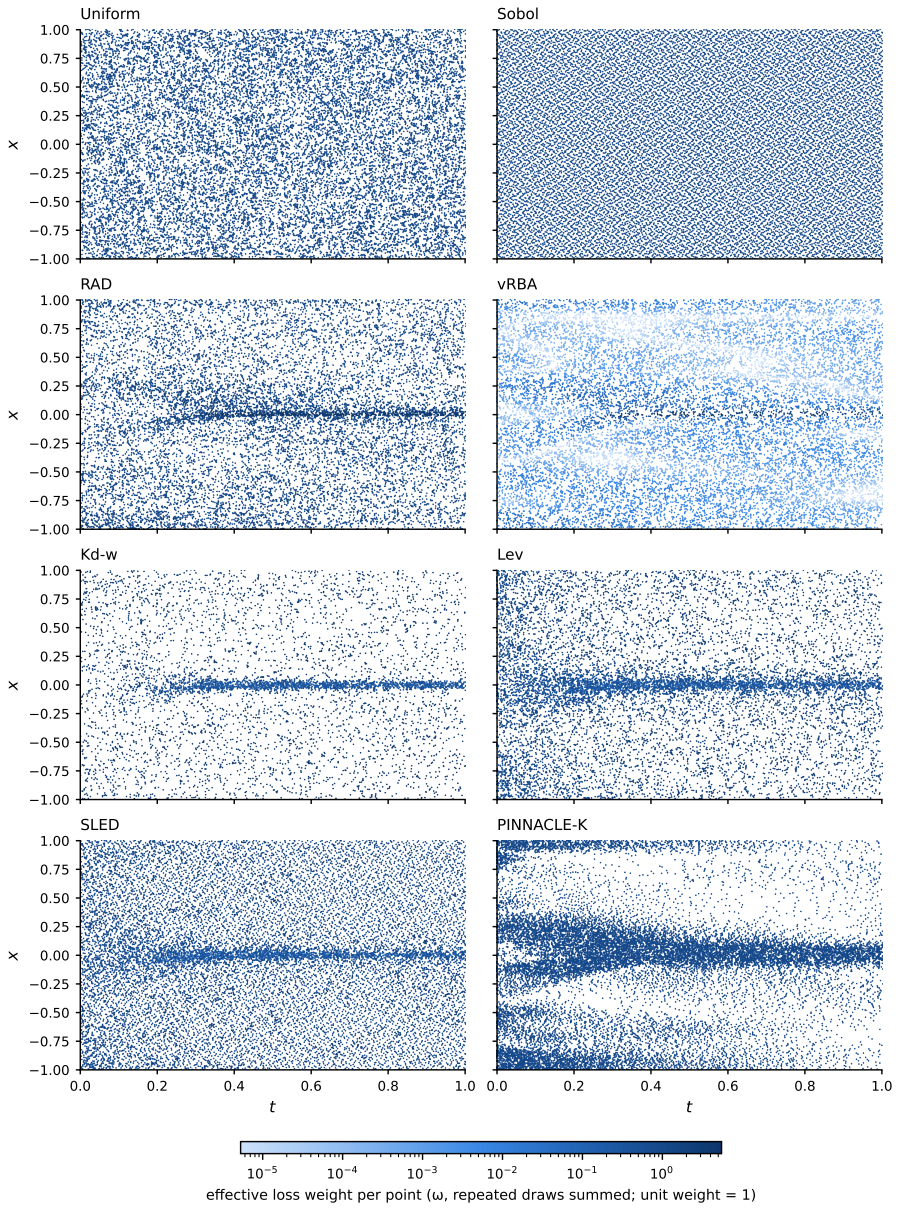

Figure 4: The batches after the warmup on Burgers, as in Figure 2. RAD, KD-W, LEV, SLED and PINNACLE-K concentrate points along the shock at $x = 0$. SLED gives $\widehat{\rho} = 0.58$ of its batch to the Sobol component.

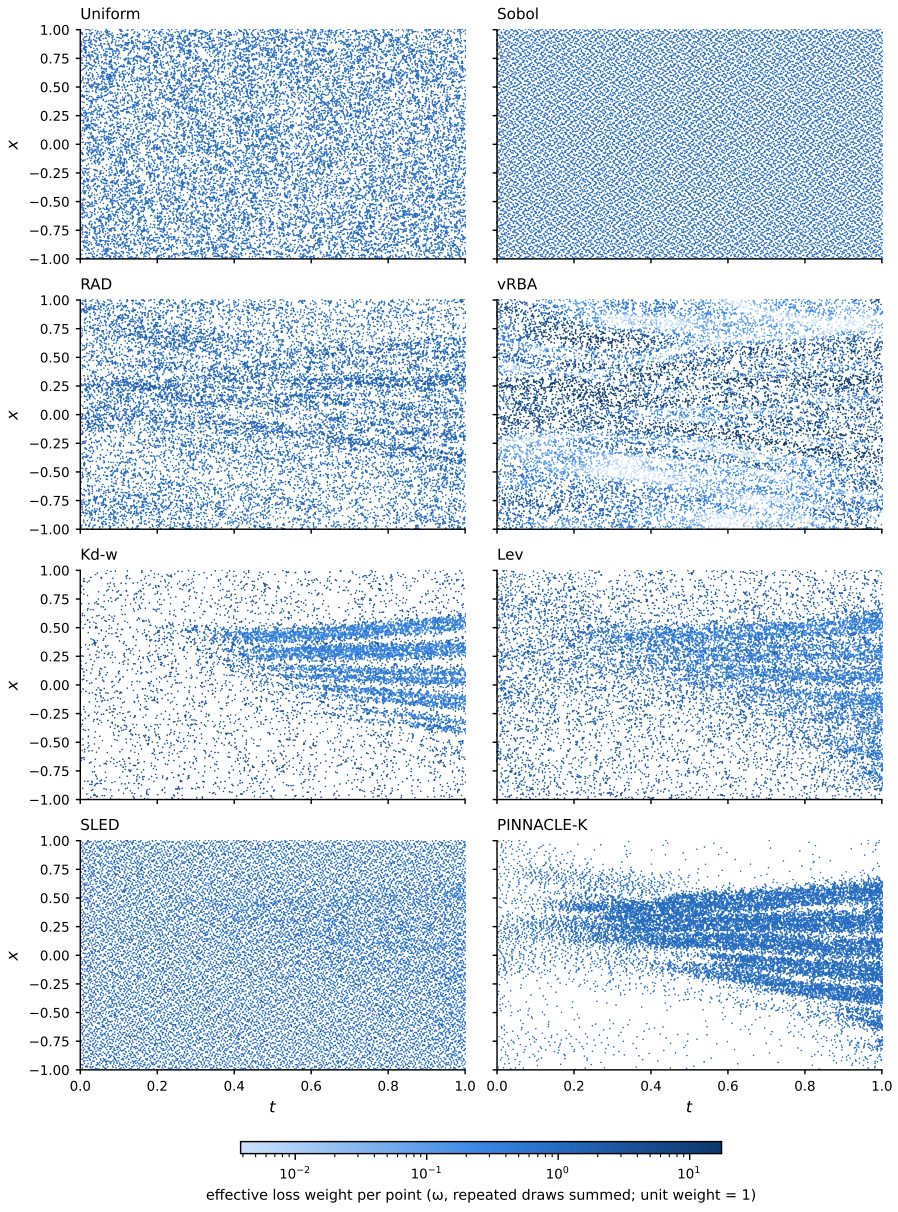

Figure 5: The batches after the warmup on KdV, as in Figure 2. KD-W, LEV and PINNACLE-K concentrate points along the paths of the solitary waves for $t \gtrsim 0.4$; SLED gives $\widehat{\rho} = 0.90$ of its batch to the Sobol component.

D LINEARIZATION DIAGNOSTICS

This appendix measures how far the second phase departs from its linearization at the warmup checkpoint θ_0, and what the linearized phase of Lemma 3.1 predicts for the frozen batches of Table 1. It covers the six frozen methods UNIF, SOBOL, RAD k=1, KD-W, LEV and SLED (RT: no KD-W run) on all six benchmarks for seeds 2, 3 and 5, that is, 105 runs, and, at the end of the phase, the 36 runs of these methods on KS and AC, whose final parameters are stored. PINNACLE-K is not covered.

Provenance. Every frozen batch is rebuilt from the warmup checkpoint of its run with the run's random-number stream. The 69 scored batches reproduce the distinct-point count, the effective sample size and the score statistics of the logged run ($5 \cdot 10^{-8}$ or better), the 36 unscored ones its checkpoint hash and distinct-point count, and the weighted sum $\sum_i \omega_i r(x_i;\theta_0)^2$ over the rebuilt rows matches the logged loss to $7 \cdot 10^{-14}$; the error after the warmup recomputed at θ_0 matches the logged value to $1.1 \cdot 10^{-15}$ (105 of 105). The off-batch loss is that of Table 2, the mean square of the residual on the same 65,536 held-out points; recomputed at the stored final parameters θ_T, it reproduces the logged final value to $2 \cdot 10^{-8}$ on 28 of the 36 KS and AC runs; on the other 8, all on AC, the last logged row lies before iteration 40,000 and the stored parameters past it, and the two values differ by up to $3 \cdot 10^{-3}$. Per-point Jacobians agree with `jax.jacrev` to 10^{-15}. Singular value decompositions are computed in float64 by a QR-based solver and checked on the KS seed-2 systems: the pool system against a Jacobi SVD (equal numerical ranks at every cutoff, the 1,000 largest singular values to $7 \cdot 10^{-11}$) and the batch system against a streaming QR factorization (equal ranks). A cutoff $\mathrm{rcond} \in \{10^{-12}, 10^{-10}, 10^{-8}\}$ keeps the singular values above $\mathrm{rcond} \cdot \sigma_{\max}$; 10^{-10} is the primary value, and the rankings below are reported at all three.

The linearized phase at θ_0 (Table 12, Figure 6). For each run, $A = W^{1/2}\Phi_C$ holds the weighted residual rows of every family (Remark A.3), and $\delta_{\mathrm{lin}} = -A^{+}W^{1/2}r_0|_C$ is the step Lemma 3.1(iii) says the linearized phase returns. Its predicted off-batch loss is the mean square of $r_0 + \Phi\delta_{\mathrm{lin}}$ on the held-out points, and its predicted error is that of $u_{\theta_0} + \nabla_\theta u_{\theta_0}^{\top}\delta_{\mathrm{lin}}$. Three findings hold on all six benchmarks. *(i) Part (iii) is the case that applies.* Every batch system is numerically rank-deficient, with a rank between 9 % (Burgers) and 28 % (KS) of $\min(\text{rows}, P)$ on the four benchmarks with one space dimension, 79 % on GL and 99.7 % on RT, and every system is inconsistent, with a relative least-squares residual on the batch of $5 \cdot 10^{-5}$ (Burgers) to $9 \cdot 10^{-2}$ (RT); so Lemma 3.1(iii), not (ii), describes the linearized phase in float64 on every run (105 of 105 at $\mathrm{rcond} = 10^{-10}$ and 10^{-8}; 103 at 10^{-12}, where two Burgers systems are consistent to 10^{-7}). *(ii) The linearized phase orders the frozen batches by their off-batch loss.* Across the methods of one seed, the predicted and the realized off-batch loss have a Spearman correlation whose median over seeds is 0.70–0.94 at $\mathrm{rcond} = 10^{-10}$, and at least 0.70 at every cutoff on every benchmark. The lowest value is on RT, where seed 2 is negative (-0.2; -0.6 at $\mathrm{rcond} = 10^{-8}$) and the other two seeds give 0.7 and 0.9; with five methods a per-seed value moves in steps of 0.1. The method predicted best is the realized best on 9 of the 18 seeds. *(iii) It predicts neither the magnitude of that loss nor the final error.* The predicted off-batch loss lies above the realized one by two to five orders of magnitude in the median on five benchmarks and by $6\times$ on RT, and above it in every run except six on KS; the realized phase reaches losses the linearized phase cannot, and $\|\delta_{\mathrm{lin}}\|_2$ is 7 to 1,500 times $\|\theta_0\|_2$, against 0.3–0.4 for the realized δ_T (Table 13). The predicted error orders the realized final errors on GL only (0.83–0.94); elsewhere the correlation is weak (KS), negative (RT), or changes sign with the cutoff (AC, Burgers, KdV), so we report no number for it. Section 5.2 uses (i) and (ii).

The network at the end of the phase (Table 13). On KS and AC, whose final parameters θ_T are stored for every method and seed, $\delta_T = \theta_T - \theta_0$ is set against the linear model at θ_0. The network leaves it. The parameters move by 0.41 (KS) and 0.33 (AC) of their norm; the linearization error $\varepsilon_{\mathrm{lin}}$ of eq. (2) at T, the root mean square of $r(\cdot;\theta_T) - r_0 - \varphi^{\top}\delta_T$ on the held-out points, is 10–11 times the residual after the warmup and $7 \cdot 10^4$ (KS) and $5 \cdot 10^5$ (AC) times the final residual; the tangent features drift by 2.0 (KS) and 0.54 (AC) of their norm and the kernel by 4.5 and 0.41. Two things persist. The leverage ranking of a uniform subsample at θ_T keeps a Spearman correlation of 0.70 (KS; per run 0.68–0.86) and 0.93 (AC; 0.89–0.97) with the ranking at θ_0, which is what scoring the batch once, at θ_0, relies on. And δ_T, although not confined to the batch row space V_C of Lemma 3.1(i), has a share of its squared norm inside V_C that is $2.0\times$ (KS) and $4.1\times$ (AC) that of

Table 12: The linearized phase at θ_0 against the realized runs, six frozen methods (RT: five), seeds 2, 3, 5, at $\mathrm{rcond} = 10^{-10}$ unless stated. Rows: residual rows of the batch system (batch points of all families and the initial-condition grid); P: parameters. Rank: numerical rank over $\min(\text{rows}, P)$, median over runs. Fit: relative least-squares residual on the batch, median. Pred./real.: predicted over realized off-batch loss, median [min, max] over runs. ρ: median over seeds of the Spearman correlation across the methods between the predicted and the realized value, at $\mathrm{rcond} = 10^{-12}\,/\,10^{-10}\,/\,10^{-8}$; error: the final relative L^2 error (RT: temperature field). Top-1: seeds on which the method predicted best is the realized best. $\|\delta_{\mathrm{lin}}\| \,/\, \|\theta_0\|$: median over runs.

	rows / P	rank	fit	pred. / real.	ρ, off-batch loss	ρ, error	top-1	$\frac{\|\delta_{\mathrm{lin}}\|_2}{\|\theta_0\|_2}$
Allen–Cahn	16,896 / 20,960	0.099	$3.3 \cdot 10^{-4}$	$1.6 \cdot 10^{4}$ [$1.1 \cdot 10^{2}$, $6.0 \cdot 10^{5}$]	0.89 / 0.94 / 0.83	0.20 / −0.26 / 0.77	1/3	36
Ginzburg–Landau	112,768 / 21,088	0.79	$8.2 \cdot 10^{-4}$	$8.8 \cdot 10^{2}$ [$1.1 \cdot 10^{2}$, $9.0 \cdot 10^{3}$]	0.89 / 0.94 / 0.89	0.89 / 0.83 / 0.94	3/3	7.2
Burgers	16,896 / 20,960	0.088	$4.6 \cdot 10^{-5}$	$1.0 \cdot 10^{5}$ [$5.4 \cdot 10^{2}$, $8.3 \cdot 10^{5}$]	0.94 / 0.89 / 0.89	−0.14 / 0.71 / 0.20	2/3	8.4
KdV	16,896 / 20,960	0.13	$2.2 \cdot 10^{-3}$	$3.9 \cdot 10^{4}$ [$4.1 \cdot 10^{3}$, $4.0 \cdot 10^{6}$]	0.77 / 0.77 / 0.77	−0.60 / −0.26 / 0.83	1/3	$3.2 \cdot 10^{2}$
Kuramoto–Siv.	16,896 / 20,961	0.28	$1.1 \cdot 10^{-4}$	$1.8 \cdot 10^{2}$ [$2.9 \cdot 10^{-6}$, $2.7 \cdot 10^{4}$]	0.77 / 0.89 / 0.94	0.26 / 0.26 / 0.03	1/3	22
Rayleigh–T.	223,840 / 21,188	0.997	$8.9 \cdot 10^{-2}$	6.3 [1.1, 31]	0.70 / 0.70 / 0.70	−0.90 / −0.90 / −0.70	1/3	$1.5 \cdot 10^{3}$

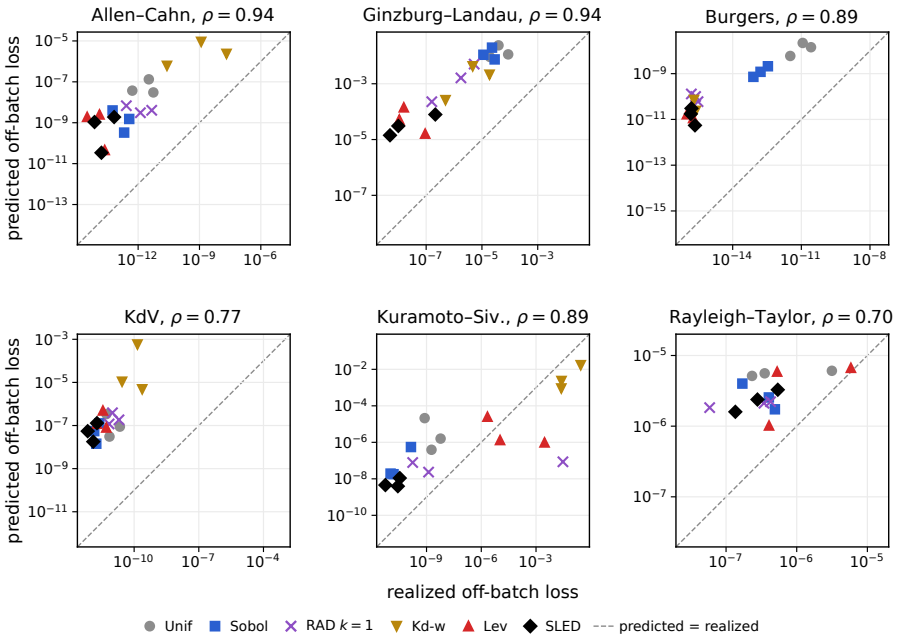

Figure 6: Predicted against realized off-batch loss, one point per run ($\mathrm{rcond} = 10^{-10}$; six frozen methods, RT five; seeds 2, 3, 5). The dashed line is equality, and ρ is the median over seeds of the Spearman correlation across the methods (Table 12). The points lie above the line, by decades on five benchmarks, and in the order of the realized loss.

a random direction; it is orthogonal to δ_{lin} at the level of a random direction ($\cos \approx 0.008$ against $1/\sqrt{P} \approx 0.007$). On AC, the eight runs whose logs end before iteration $40{,}000$ (none of them SLED) enter with their stored final parameters; over the other ten runs alone, $\varepsilon_{\mathrm{lin}}$ is 19 times the residual after the warmup and $1.4 \cdot 10^6$ times the final one, and the other medians change by less than $20\,\%$. The other four benchmarks are not measured at T.

Table 13: The network at the end of the second phase against its linearization at θ_0: medians over the six methods and three seeds (18 runs per benchmark), with the range over runs in brackets where it is wide. Drifts, alignment and leverage are evaluated on $4{,}096$ uniform points: feature drift $\|J_T - J_0\|_F \,/\, \|J_0\|_F$ with J the Jacobian of the residual, kernel drift $\|K_T - K_0\|_F \,/\, \|K_0\|_F$, alignment $\langle K_0, K_T \rangle_F / (\|K_0\|_F \, \|K_T\|_F)$, leverage: Spearman correlation of ℓ_λ at θ_0 and at θ_T. In V_C: share of $\|\delta_T\|_2^2$ inside the batch row space at θ_0 ($\mathrm{rcond} = 10^{-10}$) over that of a random direction.

	$\frac{\|\delta_T\|_2}{\|\theta_0\|_2}$	$\frac{\varepsilon_{\mathrm{lin}}}{\|r_0\|_\rho}$	$\frac{\varepsilon_{\mathrm{lin}}}{\|r(\cdot;\theta_T)\|_\rho}$	feature drift	kernel drift	alignment	leverage	in V_C	$\cos(\delta_T, \delta_{\mathrm{lin}})$
Allen–Cahn	0.33 [0.22, 0.37]	10.4 [5.7, 34]	$5.0 \cdot 10^5$	0.54	0.41 [0.23, 0.76]	0.97	0.93 [0.89, 0.97]	4.1 [3.5, 5.5]	0.0082
Kuramoto–Siv.	0.41 [0.34, 0.47]	11.1 [7.6, 12.9]	$7.3 \cdot 10^4$	2.0	4.5 [1.6, 6.3]	0.71	0.70 [0.68, 0.86]	2.0 [1.9, 2.3]	0.0085