SCALING LAWS AND MITIGATION OF MEMORIZATION IN KOLMOGOROV ARNOLD NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Malign memorization limits the deployment of Kolmogorov-Arnold Networks (KANs) under corrupted labels, particularly in scientific machine learning where observational measurements contain label corruptions. While empirical studies observe this performance degradation, existing work does not explain how the defining KAN controls, grid size G and spline degree p, govern the resulting population damage. We analyze this behavior by decomposing memorization into corruption fitting $\alpha_c(t)$ and its population footprint $V_c(t)$, giving $M_c(t) = \alpha_c(t)^2 V_c(t)$. In the localized regime of a two-layer B-spline KAN, grid refinement accelerates corruption fitting while contracting its footprint, whereas increasing degree slows fitting and broadens the footprint, yielding the matched-fit scaling $M_c = \Theta(\alpha_c^2 \sqrt{p}/G)$. This decomposition explains why fixed-budget training often report worse memorization at larger G, even though matched-fit memorization becomes strictly more localized. Guided by this analysis, we propose Curvature-Guided Basis Perturbations (CURB), which provably slows corruption fitting by perturbing curvature-sensitive basis directions during training. Across function regression, scientific operator learning (Darcy flow), and classification benchmarks, CURB reduces label corruption fitting, achieves competitive or superior accuracy to SAM and ASAM at lower per-update cost, and transfers successfully across diverse KAN basis families. In addition, CURB consistently leads trained models to flatter empirical loss landscapes.

1 INTRODUCTION

Malign memorization, where fitting corrupted labels degrades clean-data prediction, is a persistent failure mode of overparameterized models (Bayat et al., 2025; Anagnostidis et al., 2022; Zhang et al., 2017; Arpit et al., 2017). For Kolmogorov–Arnold Networks (KANs) (Liu et al., 2025), this issue is especially relevant in scientific machine learning, where target values often inherit measurement error or uncertainty (Yang et al., 2021; Zou et al., 2025).

A growing empirical literature confirms that KANs are sensitive to corrupted supervision. Target noise degrades regression recovery and classification accuracy (Shen et al., 2025b; Zeng et al., 2025; Cang et al., 2024; Zhang et al., 2025). These results establish that memorization occurs, but they do not explain how the KAN basis parameters control the malignancy of memorization. The closest theoretical perspective is the expressiveness and spectral-bias analysis of Wang et al. (2025). It explains efficient high-frequency fitting, but does not determine the population damage caused by fitting corruption.

In this paper, we make this distinction explicit by decomposing memorization as $M_c = \alpha_c^2 V_c$, where $\alpha_c(t)$ measures corruption fitting and $V_c(t)$ measures the population footprint of the fitted response. Using the KAN neural tangent kernel (Jacot et al., 2018; Wang et al., 2025), we derive how both quantities scale with (p, G). Finer grids fit corruptions faster but localize their effect more strongly, while larger spline degree slows corruption fitting but broadens the matched-fit response.

Based on our analysis, we propose a novel regularization, *Curvature-Guided Basis Perturbations* (CURB), which slows down corruption fitting. Unlike parameter-space sharpness methods such as SAM and ASAM (Foret et al., 2021; Kwon et al., 2021), or KAN-specific stochastic masking such as DropKAN (Altarabichi, 2024), CURB acts directly on the basis functions that generate KAN edge responses.

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

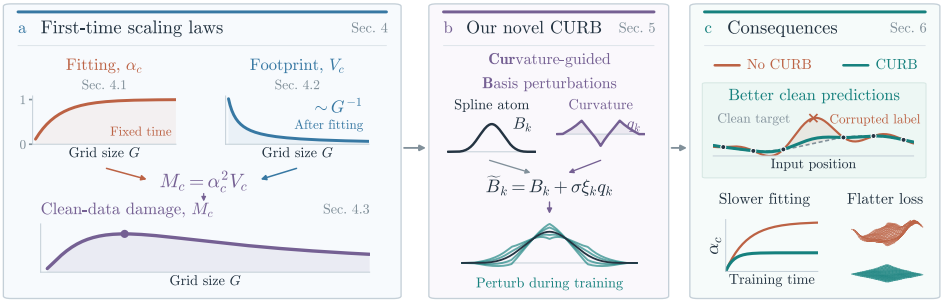

Figure 1: **From scaling laws to targeted mitigation with CURB.** Grid refinement strengthens corruption fitting, α_c, while reducing its population footprint, V_c; their product $M_c = \alpha_c^2 V_c$ explains why finer grids need not cause greater clean-data damage. **Cur**vature-guided **B**asis perturbations (CURB) reshape spline atoms during training to suppress corruption fitting and improve clean predictions, with flatter loss geometry as an additional observed benefit. Sec. 4 derives the scaling laws, Sec. 5 introduces CURB, and Sec. 6 evaluates it.

Our contributions.

- **First-time scaling laws for malign memorization in KANs.** We decompose memorization into corruption fitting $\alpha_c(t)$ and population footprint $V_c(t)$, obtaining $M_c = \Theta(\alpha_c^2 \sqrt{p}/G)$ in the localized regime and distinguishing fixed-time from matched-fit damage.
- **Novel mitigation with CURB.** CURB perturbs basis atoms along normalized curvature directions, suppressing corruption fitting without changing inference.
- **Evaluation across regression, operator learning, classification, and basis families.** Across these tasks, CURB reduces corruption fitting, is competitive with SAM and ASAM at lower update cost, transfers to Chebyshev and Gaussian bases, and produces flatter measured loss geometry.

2 Related Work

Spectral bias provides the closest existing theory of KAN optimization. Wang et al. (2025) analyze the expressiveness and spectral bias of KANs and show that KANs are less biased toward low frequencies than standard MLPs. Their analysis and experiments connect this weaker low-frequency bias with faster learning of high-frequency components, and note that the same property can make KANs more susceptible to noise and overfitting. This gives an important frequency-domain account of what KANs learn readily. Our work studies a complementary question: for a specific corrupted label, how quickly is the corruption fitted and how broadly does that fitted response affect the clean input population? The distinction is essential because grid refinement can accelerate corruption fitting while reducing the population footprint of the fitted response.

Existing noisy-label studies establish the empirical problem but leave its basis-parameter scaling unresolved. Regression studies report that target noise degrades KAN function recovery and that the severity depends on the target, sample size, and basis resolution (Shen et al., 2025a; Zeng et al., 2025; Jiang et al., 2025). Classification studies likewise report substantial clean-accuracy losses under corrupted labels (Cang et al., 2024; Pathan et al., 2025; Bar-Shalom et al., 2026). These results establish that memorization is practically important across KAN applications.
Instead, we focus on grid size and spline degree, by separating corruption fitting from population footprint, our scaling laws explain why fixed-budget comparisons across grids can differ from comparisons made at matched corruption fit.

General memorization theory motivates the distinction between fitting and population damage, while existing regularizers act on different objects. Deep networks can fit random or corrupted labels (Zhang et al., 2017; Arpit et al., 2017), and later work studies when memorization is localized, benign, or temporally separated from clean learning (Anagnostidis et al., 2022; Maini et al., 2023; Han et al., 2025; Mallinar et al., 2022; Haas et al., 2023). Robustness methods attack related failures through parameter-space perturbations, input perturbations, masking, or explicit smoothness

penalties; examples include SAM, ASAM, and DropKAN (Foret et al., 2021; Kwon et al., 2021; Altarabichi, 2024).
Instead, CURB perturbs normalized curvature directions of the functions that parameterize KAN edges.

3 Preliminaries and Memorization Setup

B-spline KANs A depth-L KAN (Liu et al., 2025) with widths $n_0, n_1, \ldots, n_L = 1$ is defined as

$$h_j^{(\ell+1)}(x) = \sum_{i=1}^{n_\ell} \phi_{ji}^{(\ell)} \left(h_i^{(\ell)}(x) \right), \qquad h^{(0)}(x) = x, \quad f_\theta(x) = h_1^{(L)}(x). \tag{1}$$

Each edge $\phi_{ji}^{(\ell)}$, connecting input node i to output node j is parameterized by a shared B-spline expansion determined by grid size G and spline degree p,

$$\phi_{ji}^{(\ell)}(z) = \sum_k a_{jik}^{(\ell)} B_{k;(p,G)}(z). \tag{2}$$

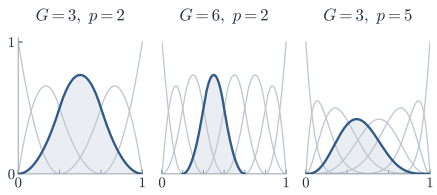

Figure 2: Increasing grid size G separates the basis functions; increasing spline degree p encourages basis function overlaps.

The basis functions are spaced by $h = 1/G$ and each occupies $p + 1$ adjacent intervals.

Corrupted labels and samples For a clean training set, $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$, let $\widetilde{\mathcal{D}}$ differ only through *corrupted labels*: in regression, a corrupted target takes the form $\widetilde{y}^{(c)} = y^{(c)} + \epsilon_c$, while in classification $y^{(c)}$ is replaced by an incorrect class label. The corresponding input sample $x^{(c)}$ is called a *corrupted sample* in the rest of the manuscript.

Memorization For a corrupted sample c, let $\Delta_c(x)$ denote the normalized prediction change attributable to its corruption. In regression,

$$\Delta_c(x) = \epsilon_c^{-1}(f_{\text{corr}}(x) - f_{\text{clean}}(x)). \tag{3}$$

For classification, Δ_c is defined analogously in the prediction associated with the corrupted label. *Memorization* is then measured as

$$M_c = \mathbb{E}_{x\sim\rho}\left[\Delta_c(x)^2\right]. \tag{4}$$

M_c measures how strongly fitting the corrupted observation changes predictions over the clean input population. Memorization is **benign** when $M_c \approx 0$, since the corrupted observation is accommodated without affecting clean inference, and **malign** when M_c remains non-negligible.

KAN neural tangent kernel Our analysis connects the spline parameters (p, G) to memorization through KAN-NTK (Jacot et al., 2018; Wang et al., 2025). The NTK for a network f_θ is

$$K_{p,G}(x, x') = \langle \nabla_\theta f_\theta(x), \nabla_\theta f_\theta(x') \rangle. \tag{5}$$

The corresponding Gram matrix on inputs $X = (x^{(1)}, \ldots, x^{(N)})$ is denoted $K_{p,G}(X, X)$. The dependence on grid size and spline degree enters through the per-edge B-spline kernel

$$k_{p,G}(z, z') = \sum_b B_{b;(p,G)}(z)\, B_{b;(p,G)}(z'), \tag{6}$$

which determines how strongly two feature coordinates share parameter-gradient directions. We also define the corresponding population kernel matrix

$$H_{p,G} = \mathbb{E}_{x\sim\rho}\left[K_{p,G}(X, x) K_{p,G}(x, X)\right]. \tag{7}$$

For MSE training in the NTK regime, let $\mathbf{e} \in \mathbb{R}^N$ denote the corruption-indicator vector, $e_i = \mathbb{1}\{i = c\}$. The normalized response to the corrupted sample satisfies

$$\Delta_c(x, t) = K_{p,G}(x, X)\, K_{p,G}(X, X)^\dagger \left(I - e^{-K_{p,G}(X,X)t/N} \right) \mathbf{e}. \tag{8}$$

Substituting this response at $t = \infty$, where the exponential decay term vanishes, into Eq. 4 gives

$$M_c = \mathbf{e}^\top K_{p,G}^\dagger H_{p,G} K_{p,G}^\dagger \mathbf{e}. \tag{9}$$

Hence the memorization induced by a corrupted sample $(x^{(c)}, y^{(c)})$ is determined by the geometry encoded by $K_{p,G}$ together with its extension to the input population encoded by $H_{p,G}$.

4 How KAN Bases Control Memorization: Scaling Laws

This section derives how grid size G and spline degree p control malign memorization in KANs. We separate memorization into the amount of corruption fitted, $\alpha_c(t)$, and the population footprint of that response, $V_c(t)$, and derive scaling laws for both. Finer grids fit corruptions faster but localize their effect more strongly, while larger spline degree slows fitting but broadens the response. For tractability, the analysis considers a two-layer B-spline KAN under MSE gradient flow.

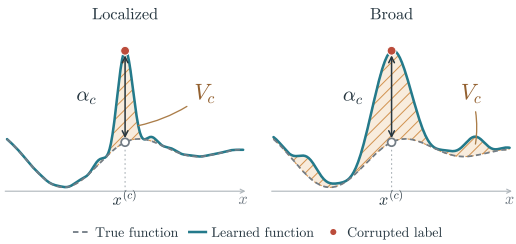

Figure 3: Two models can have the same *corruption fit* α_c while inducing very different *population footprints* V_c. Left: mostly benign memorization. Right: malign memorization.

Definition 1 (Corruption fit)**.** *For corrupted sample c, the corruption fit at training time t is* $\alpha_c(t) = \Delta_c(x^{(c)}, t)$*. It measures the fraction of the corruption reproduced by the network at* $x^{(c)}$*.*

Definition 2 (Population footprint)**.** *For* $\alpha_c(t) \neq 0$*, the population footprint of a corruption is*

$$V_c(t) = \mathbb{E}_{x\sim\rho}\left[\left(\frac{\Delta_c(x,t)}{\alpha_c(t)}\right)^2\right]. \tag{10}$$

It measures how broadly the corruption-induced response extends through the input population.

By construction, $M_c(t) = \alpha_c(t)^2 V_c(t)$. Our analysis therefore asks two questions: how do (p, G) control the fitting dynamics $\alpha_c(t)$, and how do they control the footprint $V_c(t)$? The technical derivations of the B-spline kernel $k_{p,G}$ and full KAN kernel $K_{p,G}$ are given in Appendix A in the interest of space. The following results use them to obtain the scaling laws directly.

4.1 Corruption Fitting: Grid Size Accelerates, Spline Degree Slows

The growth of $\alpha_c(t)$ depends on how the B-spline parameterization changes the tangent scale seen by the corrupted sample. Increasing G introduces increasingly localized basis functions, whereas increasing p increases their overlap and smoothness. These changes need not affect corruption fitting in the same direction. The following result gives their joint effect on the fitting dynamics.

Theorem 3 (Scaling law for corruption fitting)**.** *Consider the two-layer B-spline KAN in the localized frozen-NTK regime under MSE gradient flow. The corruption fit satisfies*

$$\alpha_c(t) = 1 - \exp\left(-\frac{\lambda_{p,G}}{N} t\right) + o(1), \tag{11}$$

where $\lambda_{p,G}$ *is the characteristic corruption-fitting rate. At fixed spline degree,* $\lambda_{p,G}$ *grows linearly with G. Moreover, in the large-degree regime* $1 \ll p \ll G$*,*

$$\lambda_{p,G} = \Theta\left(Gp^{-2} + p^{-1/2}\right). \tag{12}$$

The Gp^{-2} *and* $p^{-1/2}$ *terms arise from the first- and final-layer tangent contributions, respectively.*

Theorem 3 shows that grid refinement accelerates corruption fitting: for fixed p, its characteristic timescale decreases as G^{-1}. Increasing spline degree has the opposite effect. In the first-layer-dominated regime the fitting rate decreases as Gp^{-2}, while at sufficiently large degree the final-layer contribution scales as $p^{-1/2}$ and becomes independent of G at leading order. These trends are reflected in the corruption-fitting trajectories in Fig. 4.

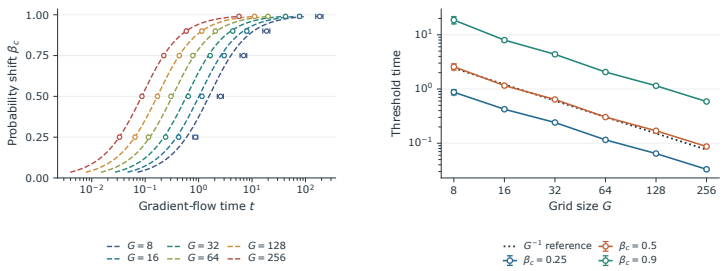

Figure 4: **Corruption-fitting time approaches inverse-grid scaling and grows with degree.** Curves show times to 25%, 50%, and 90% fitting. Dotted lines indicate the G^{-1} reference and the finite-degree prediction for 50% fitting.

4.2 Population Footprint: Grid Size Localizes, Spline Degree Broadens

The population footprint is governed by the spatial extent of the normalized KAN kernel response. Grid refinement contracts the overlap between neighboring B-spline atoms, while increasing spline degree broadens it. The following result quantifies how this basis geometry determines the population footprint of a fitted corruption.

Theorem 4 (Scaling law for population footprint)**.** *Under the assumptions of Theorem 3, suppose the first-layer tangent contribution dominates and the population density is smooth on the spline scale. In the localized regime* $1 \ll p \ll G$*, for any* t *with* $\alpha_c(t) \neq 0$*,*

$$V_c(t) = \Theta\left(\frac{\sqrt{p}}{G}\right). \tag{13}$$

Theorem 4 shows that grid refinement increasingly localizes the effect of a fitted corruption, shrinking its population footprint as G^{-1}. Increasing spline degree has the opposite geometric effect: the growing overlap between basis functions broadens the footprint as $\sqrt{p}$. Unlike the corruption-fitting dynamics, the tangent magnitude cancels under the normalization defining V_c, leaving the spatial kernel profile to determine the leading scaling (cf. Fig. 5).

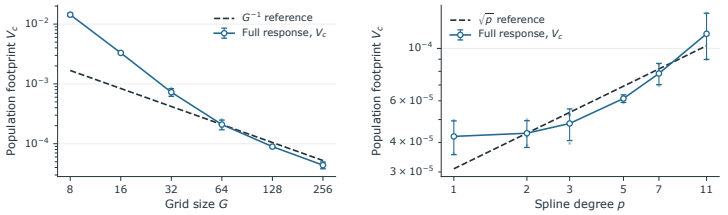

Figure 5: **The interpolated corruption footprint contracts with grid refinement and broadens with degree.** Full-Gramian MSE responses are compared with G^{-1} and $\sqrt{p}$ scaling references.

4.3 Malign or Benign? Net Scaling of Memorization

The preceding results separate two effects of the B-spline parameterization: (p, G) control how quickly the corrupted sample is fitted through $\alpha_c(t)$, and how broadly the resulting response spreads through $V_c(t)$. Combining these effects gives the scaling of malign memorization itself.

Corollary 5 (Scaling law for malign memorization)**.** *Under the assumptions of Theorems 3 and 4, in the localized regime* $1 \ll p \ll G$*,*

$$M_c(t) = \Theta\left(\alpha_c(t)^2 \frac{\sqrt{p}}{G}\right). \tag{14}$$

The corollary exposes the central tradeoff governing malign memorization. At matched corruption fit, increasing G localizes the fitted response and reduces M_c as G^{-1}, while increasing p broadens

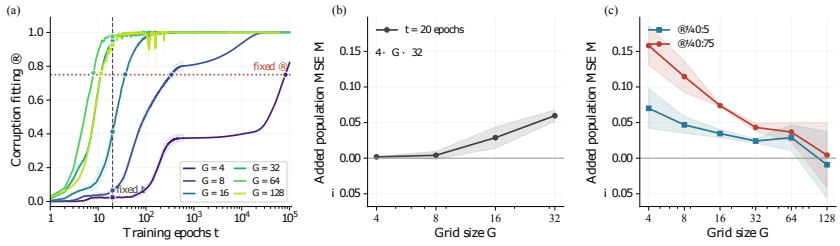

Figure 6: **Grid size changes the rate of corruption fitting, altering fixed-time comparisons.** (a) The fitting trajectories admit two comparisons: a vertical cut at a common epoch (squares), and a horizontal cut at a common corruption response (circles, $\alpha \approx 0.75$). (b) At epoch $t = 20$, added population MSE increases over the displayed range $G = 4, 8, 16, 32$, while the amount of fitted corruption also increases. (c) Comparing checkpoints at matched response levels instead reveals substantially lower added MSE at finer grids, with local nonmonotonicity at $\alpha \approx 0.5$. Results use degree-2 B-spline KANs $[8, 16, 16, 1]$, 2048 training examples, and 20% signed label corruption.

the response as $\sqrt{p}$. Larger G also accelerates the growth of $\alpha_c(t)$, so grid refinement can increase memorization early in training even though the fitted response becomes more localized.

This revises the current interpretation of KAN memorization, where larger grids have been associated with worse corrupted-label generalization (Shen et al., 2025a). At a common training time, finer grids have typically fitted more of the corruption. When checkpoints are instead matched by α_c, the induced population error decreases with grid refinement, consistent with the predicted G^{-1} scaling (cf. Fig. 6). Finer grids therefore memorize corrupted labels faster, but the resulting memorization is more localized and less malign at a matched corruption fit.

Beyond B-spline KANs. The G^{-1} law points to a broader basis-concentration mechanism. Increasing B-spline resolution concentrates the normalized basis-kernel energy and contracts the population footprint. The same behavior may arise in other KAN variants whose normalized kernel energy concentrates as the basis size increases, including globally supported constructions such as Chebyshev KANs and localized constructions such as Gaussian KANs.

5 CURB: Mitigating Malign Memorization by Slowing Corruption Fitting

We propose *Curvature-Guided Basis Perturbations (CURB)*, which perturbs curvature-sensitive basis directions during training to slow the growth of $\alpha_c(t)$. CURB target the corruption $\alpha_c(t)$ as the natural point of intervention that our analysis identified in Sec. 4 as a main contributor to malign memorization.

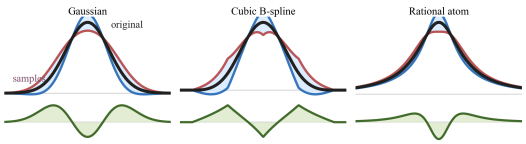

Figure 7: **CURB perturbs curvature directions, reducing memorization.** Black curves show the original basis atoms, red/blue curves show sampled CURB, and green curves show the curvature directions.

Curvature-Guided Basis Perturbations. For a B-spline atom B_k, let Π_k denote projection onto B_k under a fixed reference measure ν. CURB uses the projected curvature direction $r_k = (I - \Pi_k)B_k''$, normalized to the scale of the original atom,

$$q_k = \begin{cases} \dfrac{\|B_k\|_\nu}{\|r_k\|_\nu} r_k, & \|r_k\|_\nu > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

During training, each atom is replaced by

$$\widetilde{B}_k(z;\xi) = B_k(z) + \sigma \xi_k q_k(z), \qquad \mathcal{J}_{\mathrm{CURB}}(\theta) = \mathbb{E}_\xi[\widehat{\mathcal{L}}(f_{\theta,\xi})]. \tag{16}$$

The perturbations ξ_k are zero-mean and unit-variance, sampled independently across atoms and edges and refreshed during training. One draw is shared across the minibatch. The normalization

keeps the perturbation magnitude stable under grid refinement, while inference uses the original unperturbed basis.

Intuitively, an isolated corrupted target can be fitted through a sharp local deformation of the learned function. This spike-like interpolation of label noise also appears in analyses of benign overfitting (Mallinar et al., 2022; Haas et al., 2023). CURB perturbs the curvature directions that support such deformations. Hence, sharp corrections vary strongly under the perturbed basis and become harder to sustain, while smoother structure is less exposed to these perturbations (cf. Fig. 7).

CURB slows corruption fitting. We analyze CURB in the same local frozen-Jacobian MSE regime used in Section 4.1. Let $J \in \mathbb{R}^{N\times P}$ be the training Jacobian and let $g_c = \nabla_\theta f(x^{(c)})$ be the parameter gradient at the corrupted sample. Writing the perturbed Jacobian as $J_\xi = J + \sigma\Delta J_\xi$, with $\mathbb{E}_\xi[\Delta J_\xi] = 0$, define

$$C_{\mathrm{CURB}} = \frac{1}{N}\mathbb{E}_\xi\left[\Delta J_\xi^\top \Delta J_\xi\right] \succeq 0. \tag{17}$$

Theorem 6 (CURB slows corruption fitting)**.** *Let $\alpha_{c,\sigma}(t)$ denote the corruption fit under CURB strength σ. Under zero-mean CURB,*

$$\alpha_{c,0}(t) - \alpha_{c,\sigma}(t) = \frac{\sigma^2 t^2}{2N} g_c^\top C_{\mathrm{CURB}} g_c + O(t^3). \tag{18}$$

Hence $g_c^\top C_{\mathrm{CURB}} g_c > 0$ implies that CURB strictly slows corruption fitting for sufficiently small $t > 0$. If C_{CURB} and $J^\top J$ commute, then this ordering holds for the full local trajectory,

$$\sigma_2 \geq \sigma_1 \quad \Longrightarrow \quad \alpha_{c,\sigma_2}(t) \leq \alpha_{c,\sigma_1}(t), \qquad t \geq 0. \tag{19}$$

Theorem 6 connects CURB directly to the corruption-fitting dynamics of Section 4.1. CURB leaves the first infinitesimal response unchanged and suppresses the subsequent growth of $\alpha_c(t)$. Under spectral alignment, the slowdown persists throughout the local training trajectory.

Comparison with existing regularizers. CURB acts on the basis functions defining each KAN edge. Weight decay controls coefficient magnitude, SAM and ASAM perturb parameters, and input or feature noise perturb model inputs or representations. CURB instead regularizes sensitivity to curvature-guided basis deformations. It requires no knowledge of which samples are corrupted and introduces no modification at inference time.

6 CURB LIMITS MEMORIZATION AND IMPROVES PREDICTION

Experimental settings and further results are provided in Appendix C.

Delayed and limited corruption fitting. CURB reduces label-corruption fitting across smooth and oscillatory two-dimensional regression targets and all eight grids (Figure 8). All 24 paired runs also improve clean-test MSE. The effect extends to California Housing, where the fitted injected-offset response falls from $99.8 \pm 0.1\%$ to $6.1 \pm 0.4\%$. A complementary local-response test finds that reaching $\alpha_c = 0.25$ takes 8.42 ± 3.21 times longer under CURB (Appendix C.1). This suppression does not require sacrificing clean learning: clean-only MNIST accuracy is $92.65 \pm 0.19\%$ with CURB and $92.58 \pm 0.34\%$ with ERM.

Accurate regression under corrupted measurements. Table 1 summarizes smooth and oscillatory regression, California Housing, and Darcy operator learning. Averaging the four grids within each synthetic task, CURB reduces MSE by 40%

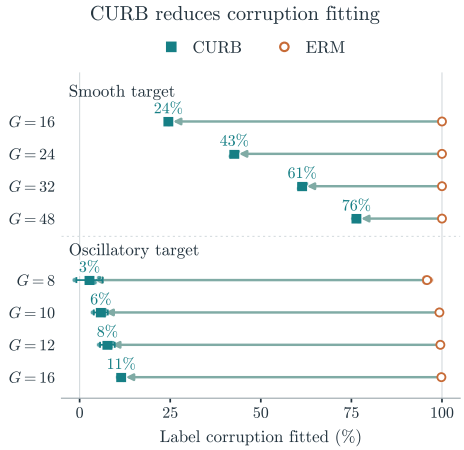

Figure 8: **CURB limits corruption fitting across tasks and grids.** Endpoint offset alignment; arrows join ERM and CURB means, with three seed points and SD bars per setting.

and 71%, respectively. SAM and ASAM perform better on the smooth target; CURB has the lowest mean error on the oscillatory target. CURB also improves over SAM by 7.9% on California and 51.9% on Darcy, where learned fields show fewer artifacts (Figure 9). Appendix C gives protocols; Appendix D.1 gives per-grid results.

Table 1: **CURB reduces regression error relative to ERM across tasks.** Test MSE, except Darcy (relative L^2, %). Synthetic tasks average four grids within each repetition; entries give mean ± SD (three repetitions). Darcy: one matched repetition. Dashes: not measured.

Task	Grids	ERM	SAM	ASAM	CURB
Smooth	4	$.0809_{\pm.0014}$	$.0232_{\pm.0016}$	$\mathbf{.0205}_{\pm.0012}$	$.0488_{\pm.0024}$
Oscillatory	4	$.1203_{\pm.0002}$	$.0604_{\pm.0018}$	$.0369_{\pm.0036}$	$\mathbf{.0346}_{\pm.0042}$
California Housing	1	$1.0564_{\pm.0149}$	$.3934_{\pm.0112}$	—	$\mathbf{.3625}_{\pm.0036}$
Darcy / AM-FNO(KAN)	1	30.10	16.89	—	**8.12**

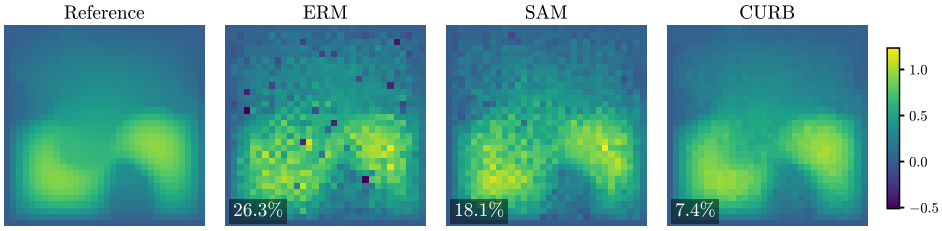

Figure 9: **CURB recovers the Darcy solution with fewer corruption artifacts.** One test field, shared colour scale; annotations give relative L^2 error. Spatial error maps appear in Appendix D.2.

Comparable accuracy across a fixed spline-coefficient budget. Across seven equal-coefficient-count KANs ($G + p = 18$), CURB averages 89.79% accuracy, close to SAM's 89.80% and above ASAM's 89.02% (Table 2). Its update cost is $1.45\times$ ERM, versus $1.97\times$ for SAM and $2.02\times$ for ASAM. In this experiment we use 10,000 MNIST images with 40% label corruption to expose a data-limited scenario to avoid suppressing memorization through number of samples available. A separate three-seed confirmation at $(G, p) = (16, 2)$ gives $89.62 \pm 0.77\%$ for CURB versus $87.74 \pm 0.46\%$ for SAM.

Table 2: **CURB approaches SAM's mean accuracy at lower update cost.** Clean MNIST test accuracy (%), final epoch 150, $[784, 96, 10]$, one repetition. All measured pairs with $G + p = 18$ are included; bold marks row maxima. Update cost is measured separately with ERM $= 1\times$ (Appendix C.2); lower is faster.

(G, p)	ERM	Input noise	DropKAN	SAM	ASAM	CURB
$(3, 15)$	73.71	82.73	77.46	**91.76**	90.43	90.60
$(6, 12)$	73.55	83.05	77.57	**90.98**	89.82	90.74
$(8, 10)$	72.23	81.62	75.94	**90.99**	90.00	90.27
$(10, 8)$	72.51	82.80	75.90	**89.67**	88.91	89.59
$(12, 6)$	72.01	83.63	77.33	**89.70**	88.34	89.01
$(14, 4)$	72.95	83.24	76.51	**89.60**	88.32	88.85
$(16, 2)$	74.33	85.68	75.12	85.90	87.29	**89.46**
Mean	73.04	83.25	76.55	**89.80**	89.02	89.79
Update cost	$1.00\times$	$0.99\times$	$1.71\times$	$1.97\times$	$2.02\times$	$1.45\times$

Alternative bases and source-only transfer. CURB improves Chebyshev/RBF MNIST accuracy and final-checkpoint USPS accuracy across all three bases (Table 3). Appendix C gives preprocessing and checkpoint sensitivity.

Table 3: **CURB transfers across bases and improves fixed-endpoint domain transfer.** Accuracy (%), three-seed mean ± SD, final epoch 150. MNIST uses 54k training images and 40% label corruption; USPS uses the same models without adaptation or target-domain selection.

	MNIST			USPS		
Basis	ERM	SAM	CURB	ERM	SAM	CURB
B-spline	$68.98_{\pm.32}$	$\mathbf{91.62}_{\pm.39}$	$91.58_{\pm2.00}$	$12.14_{\pm1.17}$	$11.54_{\pm2.01}$	$\mathbf{18.78}_{\pm1.77}$
Chebyshev	$57.41_{\pm.52}$	$85.78_{\pm.34}$	$\mathbf{95.53}_{\pm.16}$	$17.87_{\pm1.71}$	$36.21_{\pm4.00}$	$\mathbf{47.42}_{\pm5.90}$
Gaussian RBF	$58.93_{\pm1.11}$	$81.61_{\pm.27}$	$\mathbf{94.37}_{\pm.23}$	$14.15_{\pm2.20}$	$18.00_{\pm3.50}$	$\mathbf{24.83}_{\pm2.76}$

Flatter loss geometry across grid resolutions. CURB combines lower test MSE with flatter loss surfaces at all four grids (Figure 10). All-parameter probes confirm lower local sensitivity in all 12 paired runs. Further evidence on California Housing and alternative classification bases appears in Appendix C.2.

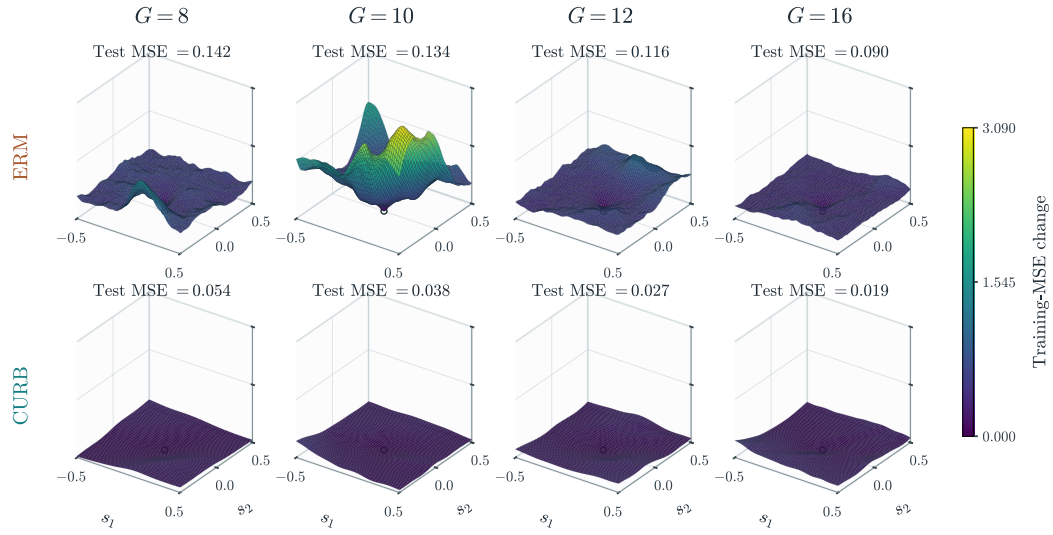

Figure 10: **CURB yields flatter loss surfaces and lower test error across grids.** ERM (top) and CURB (bottom) receive identical first-layer-bias perturbations; all surfaces share displacement, height and colour scales. Surfaces show one paired repetition; annotations report three-repetition mean clean-test MSE. Unregularized training loss is evaluated with CURB disabled.

7 Conclusion

This paper provides scaling laws for malign memorization in B-spline KANs. Our theoretical analysis shows that grid refinement accelerates corruption fitting while localizing its footprint, whereas increasing spline degree slows fitting while broadening the footprint, yielding the net matched-fit scaling $M_c = \Theta(\alpha_c^2 \sqrt{p}/G)$. This distinction explains why standard fixed-time training sweeps often overstate the malignity of finer grids, even though matched-fit memorization becomes strictly more localized. Building on this analysis, our proposed regularizer, CURB, provably delays corruption fitting by injecting perturbations along curvature-sensitive basis directions during training. Across function regression, scientific operator learning, classification, and alternative basis families, CURB suppresses corruption fitting, achieves competitive or superior generalization relative to sharpness-aware baselines at lower computational overhead, and produces flatter empirical loss surfaces.

REFERENCES

Mohammed Ghaith Altarabichi. DropKAN: Regularizing KANs by masking post-activations. *arXiv preprint arXiv:2407.13044*, 2024. URL https://arxiv.org/abs/2407.13044.

Sotiris Anagnostidis, Gregor Bachmann, Lorenzo Noci, and Thomas Hofmann. The curious case of benign memorization. *arXiv preprint arXiv:2210.14019*, 2022.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

Guy Bar-Shalom, Ami Tavory, Itay Evron, Maya Bechler-Speicher, Ido Guy, and Haggai Maron. A graph meta-network for learning on kolmogorov-arnold networks. *arXiv preprint arXiv:2602.16316*, 2026.

Reza Bayat, Mohammad Pezeshki, Elvis Dohmatob, David Lopez-Paz, and Pascal Vincent. The pitfalls of memorization: When memorization hurts generalization. In *International Conference on Learning Representations*, volume 2025, pp. 86017–86043, 2025.

Yueyang Cang, Li Shi, et al. Can kan work? exploring the potential of kolmogorov-arnold networks in computer vision. *arXiv preprint arXiv:2411.06727*, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Moritz Haas, David Holzmüller, Ulrike Luxburg, and Ingo Steinwart. Mind the spikes: Benign overfitting of kernels and neural networks in fixed dimension. *Advances in Neural Information Processing Systems*, 36:20763–20826, 2023.

Andi Han, Wei Huang, Zhanpeng Zhou, Gang Niu, Wuyang Chen, Junchi Yan, Akiko Takeda, and Taiji Suzuki. On the role of label noise in the feature learning process. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21733–21773. PMLR, 2025. URL https://proceedings.mlr.press/v267/han25c.html.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Jiun-Cheng Jiang, Morris Yu-Chao Huang, Tianlong Chen, and Hsi-Sheng Goan. Quantum variational activation functions empower kolmogorov-arnold networks. *arXiv preprint arXiv:2509.14026*, 2025.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5905–5914. PMLR, 2021. URL https://proceedings.mlr.press/v139/kwon21b.html.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, volume 2025, pp. 70367–70413, 2025.

Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23536–23557. PMLR, 2023. URL https://proceedings.mlr.press/v202/maini23a.html.

Neil Mallinar, James Simon, Amirhesam Abedsoltan, Parthe Pandit, Misha Belkin, and Preetum Nakkiran. Benign, tempered, or catastrophic: Toward a refined taxonomy of overfitting. *Advances in neural information processing systems*, 35:1182–1195, 2022.

Arif Khan Pathan, Raghavendra Kulkarni, Subbareddy Batreddy, Giridhar Pamisetty, Priya Verma, and Ch Sobhan Babu. Robust graph classification with instance independent noisy labels: An em approach using kolmogorov-arnold networks. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2025.

Haoran Shen, Chen Zeng, Jiahui Wang, and Qiao Wang. Reduced effectiveness of Kolmogorov-Arnold Networks on functions with noise. In *ICASSP 2025 – 2025 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2025a. URL https://arxiv.org/abs/2407.14882.

Haoran Shen, Chen Zeng, Jiahui Wang, and Qiao Wang. Reduced effectiveness of kolmogorov-arnold networks on functions with noise. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025b.

Yixuan Wang, Jonathan W. Siegel, Ziming Liu, and Thomas Y. Hou. On the expressiveness and spectral bias of KANs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/448444518637da106d978ae7409d9789-Abstract-Conference.html.

Zipeng Xiao, Siqi Kou, Zhongkai Hao, Bokai Lin, and Zhijie Deng. Amortized fourier neural operators. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d06a797c436cd5136a6f45b063316278-Abstract-Conference.html.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Chen Zeng, Jiahui Wang, Haoran Shen, and Qiao Wang. Kan versus mlp on irregular or noisy function. In *2025 15th IEEE International Conference on Pattern Recognition Systems (ICPRS)*, pp. 1–7. IEEE, 2025.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.03530.

Xinlei Zhang, Fan Liu, Chuanyi Zhang, Fan Cheng, Qian Li, and Yuhui Zheng. Towards label-noise resistant learning via optimal brain damage masking. *arXiv preprint arXiv:2508.09697*, 2025.

Zongren Zou, Xuhui Meng, and George Em Karniadakis. Uncertainty quantification for noisy inputs–outputs in physics-informed neural networks and neural operators. *Computer Methods in Applied Mechanics and Engineering*, 433:117479, 2025.

A Kernel Lemmas

Lemma 7 (B-spline kernel geometry)**.** *Let*

$$k_{p,G}(z,z') = \sum_b \psi_{b;(p,G)}(z)\psi_{b;(p,G)}(z')$$

be the kernel induced by an interior uniform degree-p B-spline basis with grid size G. Then

$$|z-z'| > \frac{p+1}{G} \quad \Longrightarrow \quad k_{p,G}(z,z') = 0.$$

Averaging over the position of the points relative to the spline grid gives

$$\mathbb{E}[k_{p,G}(z,z')] = B_{2p+1}(p+1+G(z'-z))\,, \tag{20}$$

and

$$\mathbb{E}[\partial_z \partial_{z'} k_{p,G}(z,z')] = -G^2 B''_{2p+1}(p+1+G(z'-z))\,. \tag{21}$$

In particular, the spline-coupling width is of order $\sqrt{p+1}/G$.

Proof. Away from the open boundaries of the spline domain, the basis functions are translations of the degree-p cardinal B-spline:

$$\psi_{b;(p,G)}(z) = B_p(Gz - b),$$

up to an irrelevant indexing shift. Hence

$$k_{p,G}(z, z') = \sum_b B_p(Gz - b)B_p(Gz' - b).$$

Support. The cardinal spline B_p has support of length $p + 1$. A summand can therefore be nonzero only if the same translated spline contains both z and z'. Thus

$$G|z - z'| \le p + 1$$

is necessary for nonzero coupling, which proves the support statement.

Averaged kernel. Let the common position of the two points relative to the grid vary uniformly over one grid interval while keeping $z' - z$ fixed. Averaging over that common position converts the sum over translates into the autocorrelation

$$\mathbb{E}[k_{p,G}(z, z')] = \int_{\mathbb{R}} B_p(u)B_p(u + G(z' - z))\, du.$$

The cardinal B-spline convolution identity gives

$$\int_{\mathbb{R}} B_p(u)B_p(u + s)\, du = B_{2p+1}(p + 1 + s),$$

so with $s = G(z' - z)$,

$$\mathbb{E}[k_{p,G}(z, z')] = B_{2p+1}(p + 1 + G(z' - z)),$$

which is equation 20.

Derivative kernel. Differentiating the averaged kernel with respect to both arguments gives

$$\partial_z \partial_{z'} B_{2p+1}(p + 1 + G(z' - z)) = -G^2 B''_{2p+1}(p + 1 + G(z' - z)),$$

which proves equation 21.

Width. The centered profile in equation 20 is a degree-$(2p + 1)$ cardinal B-spline. Its variance is $(p + 1)/6$. After the rescaling $s = G(z' - z)$, the variance in the original coordinate is therefore

$$\frac{p + 1}{6G^2},$$

so the corresponding root-mean-square coupling width is

$$\frac{\sqrt{p + 1}}{\sqrt{6}\, G}.$$

Hence the coupling width is of order $\sqrt{p + 1}/G$. □

Lemma 8 (Two-layer KAN tangent kernel)**.** *For the two-layer KAN*

$$h_j(x) = \sum_{i,b} w^{(1)}_{ji,b} \psi_{b;(p,G)}(x_i), \qquad f(x) = \sum_{j,b} w^{(2)}_{j,b} \psi_{b;(p,G)}(h_j(x)),$$

define

$$D_j(x) = \sum_b w^{(2)}_{j,b} \psi'_{b;(p,G)}(h_j(x)).$$

Its NTK is

$$K_{p,G}(x, x') = \sum_{j=1}^{m} k_{p,G}(h_j(x), h_j(x')) + \left[\sum_{i=1}^{d_0} k_{p,G}(x_i, x'_i)\right] \left[\sum_{j=1}^{m} D_j(x)D_j(x')\right]. \tag{22}$$

Assume the second-layer spline coefficients are independent, mean zero, and satisfy

$$\mathbb{E}\left[(w^{(2)}_{j,b})^2\right] = \frac{\sigma_2^2}{m(G + p)}.$$

If the input and hidden coordinates remain in the interior spline region and their locations relative to the knot grid are averaged as in Lemma 1, then

$$K_{p,G}(x, x) = mB_{2p+1}(p + 1) + \frac{\sigma_2^2 d_0 G^2}{G + p} B_{2p+1}(p + 1)\left[-B''_{2p+1}(p + 1)\right] + o(1). \tag{23}$$

Proof. The NTK is the sum of gradient inner products over all trainable spline coefficients.

Second-layer coefficients. For a second-layer coefficient,

$$\frac{\partial f(x)}{\partial w_{j,b}^{(2)}} = \psi_{b;(p,G)}(h_j(x)).$$

Therefore the contribution from all second-layer coefficients is

$$\sum_{j,b} \psi_{b;(p,G)}(h_j(x))\psi_{b;(p,G)}(h_j(x')) = \sum_{j=1}^{m} k_{p,G}(h_j(x), h_j(x')).$$

First-layer coefficients. By the chain rule,

$$\frac{\partial f(x)}{\partial w_{ji,b}^{(1)}} = D_j(x)\psi_{b;(p,G)}(x_i).$$

Hence the contribution from all first-layer coefficients is

$$\sum_{j,i,b} D_j(x)D_j(x')\psi_{b;(p,G)}(x_i)\psi_{b;(p,G)}(x_i').$$

Separating the sums gives

$$\left[\sum_{i=1}^{d_0} k_{p,G}(x_i, x_i')\right]\left[\sum_{j=1}^{m} D_j(x)D_j(x')\right].$$

Adding the first- and second-layer contributions proves equation 22.

Second-layer diagonal contribution. When we set $x' = x$, by Lemma 1,

$$\mathbb{E}[k_{p,G}(h_j(x), h_j(x))] = B_{2p+1}(p+1).$$

Summing over j gives

$$\mathbb{E}\Big[K_{p,G}^{(2)}(x,x)\Big] = mB_{2p+1}(p+1).$$

First-layer sensitivity scale. For each hidden unit,

$$D_j(x) = \sum_b w_{j,b}^{(2)}\psi'_{b;(p,G)}(h_j(x)).$$

Using independence and zero mean of the second-layer coefficients,

$$\mathbb{E}[D_j(x)^2] = \frac{\sigma_2^2}{m(G+p)}\sum_b \psi'_{b;(p,G)}(h_j(x))^2.$$

Lemma 1 gives

$$\mathbb{E}\left[\sum_b \psi'_{b;(p,G)}(h_j(x))^2\right] = -G^2 B''_{2p+1}(p+1).$$

Thus

$$\mathbb{E}[D_j(x)^2] = \frac{\sigma_2^2 G^2}{m(G+p)}\left[-B''_{2p+1}(p+1)\right].$$

Summing over the m hidden units,

$$\mathbb{E}\left[\sum_{j=1}^{m} D_j(x)^2\right] = \frac{\sigma_2^2 G^2}{G+p}\left[-B''_{2p+1}(p+1)\right].$$

Input spline factor. Again by Lemma 1,

$$\mathbb{E}\left[\sum_{i=1}^{d_0} k_{p,G}(x_i, x_i)\right] = d_0 B_{2p+1}(p+1).$$

Multiplying this factor by the sensitivity factor gives the first-layer diagonal contribution

$$\frac{\sigma_2^2 d_0 G^2}{G+p} B_{2p+1}(p+1)\left[-B''_{2p+1}(p+1)\right].$$

Adding the first- and second-layer diagonal terms yields equation 23. The $o(1)$ term records the boundary and finite-sample deviations excluded by the interior averaging assumptions. □

B Proofs for Theorems 3, 4,6

B.1 Proof of Theorem 3

Proof. MSE NTK dynamics and Localized training Gramian. For a single corrupted label, frozen-NTK gradient flow gives

$$\alpha_c(t) = e_c^\top \left(I - e^{-K_{p,G}t/n}\right) e_c.$$

In the localized regime, the off-diagonal entries of the training Gramian are lower order than its diagonal contribution. Therefore the effective eigenvalue seen by sample c is asymptotically K_{cc}, and

$$\alpha_c(t) = 1 - \exp\left(-\frac{K_{cc}}{n}t\right) + o(1).$$

NTK diagonal. By Lemma 2,

$$K_{cc} = mB_{2p+1}(p+1) + \frac{\sigma_2^2 d_0 G^2}{G+p} B_{2p+1}(p+1)\left[-B''_{2p+1}(p+1)\right] + o(1).$$

Substitution yields Eq. 12.

Grid asymptotics. For fixed p,

$$\frac{G^2}{G+p} \sim G,$$

so the first-layer contribution grows linearly with G and the fitting rate is proportional to G.

Degree asymptotics. For large p,

$$B_{2p+1}(p+1) \sim \sqrt{\frac{3}{\pi p}},$$

and

$$-B''_{2p+1}(p+1) \sim 6\sqrt{\frac{3}{\pi}}\, p^{-3/2}.$$

Hence, when $1 \ll p \ll G$ and the first-layer contribution dominates,

$$\frac{G^2}{G+p} B_{2p+1}(p+1)\left[-B''_{2p+1}(p+1)\right] = \Theta(Gp^{-2}).$$

If instead the final-layer contribution dominates at larger degree, its scale is

$$mB_{2p+1}(p+1) = \Theta(p^{-1/2}).$$

This proves the stated grid and degree scaling. □

B.2 Proof of Theorem 4

Proof. Converged MSE response. Once the training Gramian is full rank,

$$\alpha_c(\infty) = 1,$$

and the normalized corruption response is

$$V_c = \mathbb{E}_{x\sim\rho}\left[\left(K_{p,G}(x,X)K_{p,G}^{-1}e_c\right)^2\right].$$

Localized inverse. In the same localized regime as Theorem 1,

$$K_{p,G}^{-1} e_c = \frac{e_c}{K_{cc}} + o(K_{cc}^{-1}),$$

so

$$V_c = \mathbb{E}_{x\sim\rho}\left[\left(\frac{K_{p,G}(x, x_c)}{K_{cc}}\right)^2\right] + o(1).$$

First-layer normalization. Under first-layer dominance, Eq. 22 gives

$$K_{p,G}(x, x_c) \approx \left[\sum_{i=1}^{d_0} k_{p,G}(x_i, x_{c,i})\right]\left[\sum_{j=1}^{m} D_j(x) D_j(x_c)\right].$$

The same derivative scale appears in K_{cc}. After normalization, the $G^2/(G+p)$ derivative magnitude cancels, so the leading p, G dependence is carried by the normalized spline overlap.

Grid scaling. Using the cardinal representation

$$\psi_{b;(p,G)}(z) = B_p(Gz - b),$$

write the relative coordinate displacement as $s = G(z' - z)$. The population integral therefore contributes a Jacobian factor G^{-1}, yielding

$$V_c = \Theta\left[\frac{1}{G}\int_0^1 \int_{\mathbb{R}} \left(\frac{\sum_b B_p(u + s - b) B_p(u - b)}{\sum_b B_p(u - b)^2}\right)^2 ds\, du\right].$$

This proves Eq. 13 and, for fixed p,

$$V_c = \Theta(G^{-1}).$$

Degree scaling. For large p, the normalized cardinal B-spline overlap approaches the corresponding Gaussian autocorrelation profile, giving

$$\int_0^1 \int_{\mathbb{R}} \left(\frac{\sum_b B_p(u + s - b) B_p(u - b)}{\sum_b B_p(u - b)^2}\right)^2 ds\, du \sim \sqrt{\frac{\pi p}{6}}.$$

Therefore

$$V_c = \Theta\left(\frac{\sqrt{p}}{G}\right)$$

within the localized first-layer-dominant regime. □

B.3 Proof of Theorem 6

Proof. Let $s \in \mathbb{R}^N$ denote the unit corruption response on the training set, with $s_i = \mathbf{1}\{i = c\}$. Then $J^\top s = g_c$. Let $u_\sigma(t)$ denote the parameter response per unit label corruption, initialized at $u_\sigma(0) = 0$. Under the local frozen-Jacobian model, the expected CURB objective is

$$\begin{aligned}\mathbb{E}_\xi L_\xi(u) &= \frac{1}{2N}\mathbb{E}_\xi \left\|(J + \sigma \Delta J_\xi)u - s\right\|^2 \\ &= \frac{1}{2} u^\top \left(\frac{J^\top J}{N} + \sigma^2 C_{\mathrm{CURB}}\right) u - \frac{1}{N} g_c^\top u + \text{const.}\end{aligned} \tag{24}$$

The mixed terms vanish because $\mathbb{E}_\xi[\Delta J_\xi] = 0$. Define

$$A_\sigma = \frac{J^\top J}{N} + \sigma^2 C_{\mathrm{CURB}}.$$

Gradient flow therefore satisfies

$$\dot{u}_\sigma(t) = -A_\sigma u_\sigma(t) + \frac{g_c}{N}.$$

Expanding around $t = 0$ gives

$$u_\sigma(t) = \frac{t}{N} g_c - \frac{t^2}{2N} A_\sigma g_c + O(t^3).$$

Since $\alpha_{c,\sigma}(t) = g_c^\top u_\sigma(t)$,

$$\alpha_{c,\sigma}(t) = \frac{t}{N}\|g_c\|^2 - \frac{t^2}{2N} g_c^\top \left(\frac{J^\top J}{N} + \sigma^2 C_{\mathrm{CURB}} \right) g_c + O(t^3).$$

Subtracting the $\sigma = 0$ trajectory yields

$$\alpha_{c,0}(t) - \alpha_{c,\sigma}(t) = \frac{\sigma^2 t^2}{2N} g_c^\top C_{\mathrm{CURB}} g_c + O(t^3),$$

which proves Eq. equation 18.

For the all-time result, suppose $C_{\mathrm{CURB}} J^\top J = J^\top J C_{\mathrm{CURB}}$. The two symmetric operators then admit a common orthonormal eigenbasis $\{q_r\}$ satisfying

$$\frac{J^\top J}{N} q_r = h_r q_r, \qquad C_{\mathrm{CURB}} q_r = c_r q_r, \qquad h_r, c_r \geq 0.$$

Writing $g_c = \sum_r \beta_r q_r$, the solution of the linear system gives

$$\alpha_{c,\sigma}(t) = \frac{1}{N} \sum_r \beta_r^2 \frac{1 - e^{-(h_r + \sigma^2 c_r)t}}{h_r + \sigma^2 c_r},$$

with the continuous value t when the denominator vanishes. For $a \geq 0$, the function

$$\phi_t(a) = \frac{1 - e^{-at}}{a}$$

is nonincreasing in a. Since every $c_r \geq 0$, each term is nonincreasing in σ^2. Therefore

$$\sigma_2 \geq \sigma_1 \quad \Longrightarrow \quad \alpha_{c,\sigma_2}(t) \leq \alpha_{c,\sigma_1}(t)$$

for every $t \geq 0$, proving Eq. equation 19. □

C Experimental settings and further results

Evaluation protocol. We compare final checkpoints at matched training budgets to measure robustness to continued learning from corrupted labels. Validation-selected checkpoint results are reported separately. Error bars and ± values denote standard deviations across independent repetitions; configurations are averaged within each repetition before computing task-level uncertainty.

MNIST: data-limited configuration comparison. The experiment uses 9,000 training, 900 validation, and 10,000 official test images; 40% of training labels are changed to incorrect classes, while test labels remain clean. The data-limited setting exposes memorization under limited clean constraints. Models use $[784, 96, 10]$ B-spline edges with LayerNorm, batch size 1,024, AdamW learning rate 0.003, weight decay 10^{-4}, and 150 epochs. The seven main-table configurations satisfy $G + p = 18$, holding the number of spline coefficients per edge fixed while varying locality and smoothness.

CURB uses strength 1.75 with transport-projected, atom-normalized, shared-input perturbations on every update. SAM uses radius 3, selected by a seven-trial, 150-epoch clean-validation anchor search at $(G, p) = (10, 8)$. ASAM uses radius 5 after a full-horizon strength screen. Input noise uses validation-selected standard deviation 1.5 and DropKAN uses probability 0.2. Regularizer strengths are selected with the learning rate and weight decay held fixed.

Input-noise strength selection. Input-noise strength is selected at $(G, p) = (10, 8)$ using clean validation after 150 epochs. The search covers standard deviations $\{0, 0.3, 0.6, 1, 1.5, 3, 6\}$ in normalized-input units, with zero corresponding to ERM. The selected strength, 1.5, gives 82.80% test accuracy versus 72.51% for ERM. Perturbed inputs are clamped to the valid pixel range; evaluation uses unperturbed images. The selected strength is then fixed across all seven main-table configurations. The table below reports the strength sweep.

Table 4: **Tuned input augmentation improves robustness to label noise.** MNIST accuracy (%) after 150 epochs at $(G, p) = (10, 8)$. Bold identifies the strength selected by clean validation.

Input std.	0	0.3	0.6	1	1.5	3	6
Validation	70.11	70.22	69.00	77.44	**83.44**	79.00	64.67
Test	72.51	69.03	68.02	75.72	**82.80**	80.59	66.96

Baselines. The long-horizon comparison includes SAM (Foret et al., 2021), ASAM (Kwon et al., 2021), input augmentation, and DropKAN (Altarabichi, 2024), covering sharpness-aware optimization, generic input perturbation, and KAN-specific stochastic regularization.

Cross-basis MNIST and USPS. All three model families use $[784, 96, 10]$, with B-spline $p = 2, G = 16$, Chebyshev degree 5, or seven Gaussian RBF atoms. The Gaussian model uses a FastKAN-style RBF basis. The full split is 54k training and 6k clean validation images, with 10k official test images, 40% forced-wrong training labels, and 150 epochs. Chebyshev/Gaussian SAM radii $\{0.05, 0.1, 0.3, 1, 3\}$ were screened on a separate 9k/900 development split for 100 epochs, selecting the final clean-validation accuracy (CE breaks ties). Chosen radii are 3 and 1, respectively; these strengths and optimizer settings are then held fixed on the full training split.

California Housing. Models are $[8, 64, 64, 1]$ B-spline KANs with $p = 2, G = 16$, trained for 500 epochs with signed two-target-standard-deviation outliers at 20% of training labels; evaluation labels remain clean. Three repetitions use matched data and initialization. Development validation at the full horizon selects CURB strength 0.6 and the SAM setting. Validation-selected test MSE also favours CURB (0.3637) over SAM (0.3973). The fitted-offset statistic projects the prediction difference between paired noisy and clean training onto the known offsets at corrupted observations. It summarizes simultaneous corruptions, not isolated-point α_c.

Darcy scientific operator learning. The official NeuralOperator 32×32 Darcy data[1] use 1,000 training, 200 validation, and 1,000 test fields. Corruption changes 20% of interior scalar output observations by signed two-target-standard-deviation offsets; inputs and evaluation targets stay clean. AM-FNO(KAN) (Xiao et al., 2024) uses four blocks, width 16, B-spline $G = 32, p = 3$ filters, and an identity base branch. Training minimizes mean relative L^2 error for 500 epochs. CURB strengths $\{0.1, 0.3, 0.5, 1, 3, 10\}$ and SAM radii $\{0.01, 0.05, 0.2, 1\}$ are compared at that horizon, selecting strength 1 and radius 0.2 by validation.

Corruption response and loss geometry. The trajectory test uses a two-layer width-12, two-input, $G = 16, p = 2$ model trained for 1,000 AdamW updates on 64 observations with 25% signed ± 0.5 corruption. Paired ± 0.01 single-label probes start from each method's own checkpoint and continue for 2,000 full-batch GD updates at learning rate 0.01. The controlled CURB objective integrates its noise analytically through an exposure term. These method-specific starting checkpoints characterize the response of the learned solutions. The larger endpoint sweeps use stochastic all-layer CURB with $[2, 24, 24, 1]$, $p = 2$, SiLU base branches, 20% signed ± 0.75 offsets, 600 AdamW epochs, and CURB strength 1. The smooth and oscillatory grid ranges are $\{16, 24, 32, 48\}$ and $\{8, 10, 12, 16\}$. Each task uses three independent repetitions, with paired data and initialization.

Figures 11&12 demonstrate the loss landscape for networks trained without (top) and with (bottom) CURB. Beyond improving clean-test performance, CURB favors flatter loss geometry. Across the grid sizes examined, the no-CURB solutions exhibit steeper loss surfaces, whereas the CURB solutions tolerate substantially larger weight perturbations with little change in loss. This contrast, together with the consistently lower test error, highlights the additional benefit of reduced sensitivity of the training loss to parameter perturbations.

SAM and ASAM on smooth and oscillatory regression. With $u = (x + 1)/2$ for $x \in [-1, 1]^2$, the two targets are

$$f_q(x) = 0.5 \sin(2\pi q u_1) + 0.25 \cos(4\pi q u_2) + 0.15 \sin\big(2\pi q (u_1 + u_2)\big),$$

[1]NeuralOperator Team, *Darcy Flow Dataset*, version 2, Zenodo. https://doi.org/10.5281/zenodo.12784353.

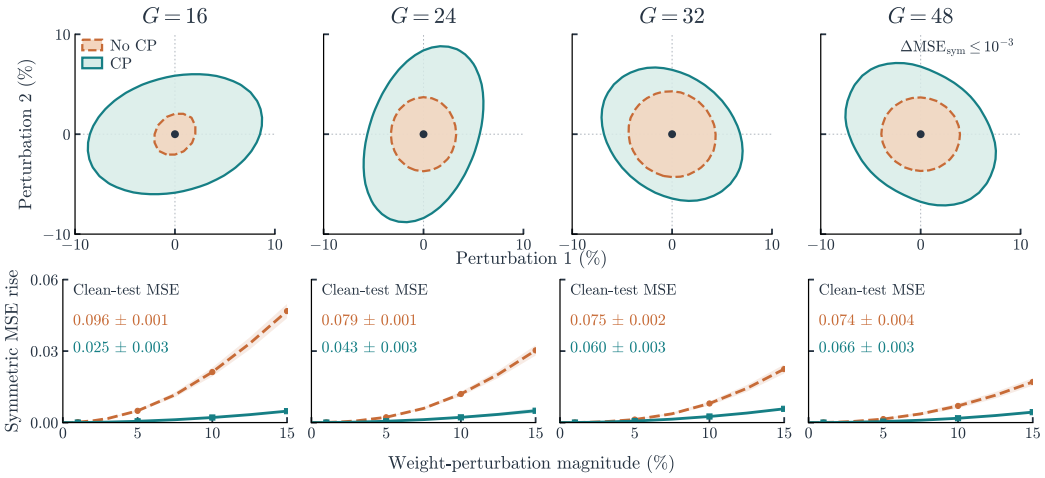

Figure 11: CURB reduces sensitivity to weight perturbations while improving clean-test MSE across grid sizes. A $[2, 24, 24, 1]$ B-spline KAN ($p = 2$, SiLU branch enabled) is trained on 512 samples of a smooth two-dimensional regression target with 20% corrupted labels. Upper panels show low-sensitivity regions in a common two-dimensional weight-perturbation plane for seed 700. For a displacement δ, the plotted rise is $\Delta\mathcal{L} = [\mathcal{L}(\theta + \delta) + \mathcal{L}(\theta - \delta)]/2 - \mathcal{L}(\theta)$; the contours use the same threshold 10^{-3} in all panels. Perturbations have identical directions and absolute amplitudes within each paired comparison, normalized by the pair-average parameter norm of each layer. Lower panels show the mean response over 24 random directions and three independent paired seeds, with one-standard-deviation bands across seeds; colored numbers give mean ± standard deviation of clean-test MSE. The same noisy-training MSE is evaluated for both methods with CURB switched off.

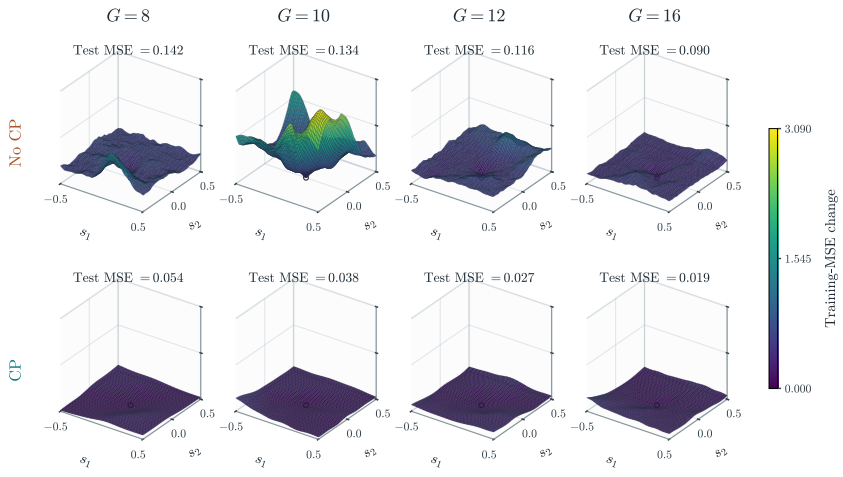

Figure 12: CURB produces flatter weight-space loss surfaces and lower clean-test error across grid sizes. No-CURB (top) and CURB (bottom) checkpoints are perturbed along the same two random weight directions. Height and color show $\mathcal{L}(\theta + s_1 d_1 + s_2 d_2) - \mathcal{L}(\theta)$, using the unregularized noisy-training MSE. All eight panels share axis limits, color scale, and viewing angle. Panel annotations report clean-test MSE at the unperturbed checkpoint.

using $q = 1$ (smooth) and $q = 3$ (oscillatory). Scrambled Sobol inputs provide 512 and 1,024 training observations, respectively, with independent sets of 1,024 validation and 8,192 test points for each task. Both baselines use the same data, initial parameters, minibatch order and 600-epoch budget as the ERM/CURB comparisons. AdamW uses learning rate 0.0015, zero weight decay,

batch size 64 and cosine decay to 2% of the initial rate; gradients are clipped to norm 10 before the final update. An independent development repetition selects one radius per task and method by mean final validation MSE across the four grids. SAM tests $\{0.01, 0.05, 0.2, 1, 3\}$; ASAM tests $\{0.05, 0.2, 1, 3, 10\}$ with adaptive offset 0.01. Selected radii are 1 for both methods on the smooth task, and 0.2 (SAM) and 1 (ASAM) on the oscillatory task. All selections are interior to the search range. Each selected setting is evaluated in three paired repetitions, giving 48 baseline runs.

C.1 Corruption-response trajectories

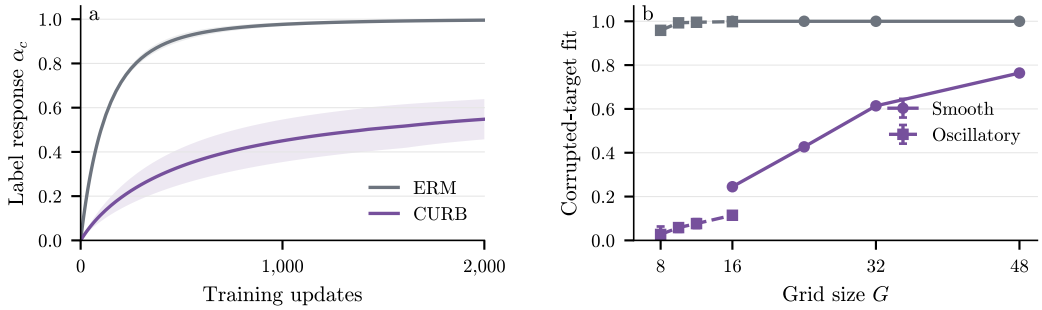

Figure 13: **CURB delays local label response and limits final offset fitting.** Left: paired single-label response, mean and seed SD over three seeds. Right: endpoint offset alignment across the eight task/grid settings, with seed-SD bars. These are different experiments and response measures; their protocols are specified above.

C.2 Loss-geometry protocol and cross-grid confirmation

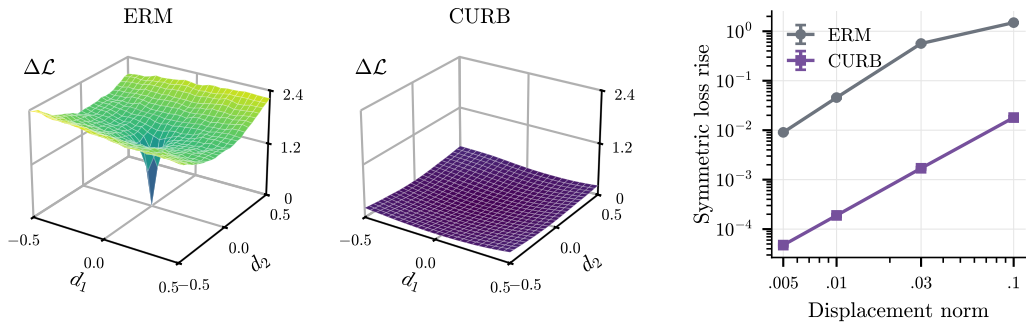

Figure 14: **CURB's lower loss sensitivity extends to California Housing.** Matched first-hidden-bias slices share their axes and color scale. Right: an independent all-parameter perturbation test, with 32 directions per seed and three seeds; bars show SD across seed means.

California Housing and classification. Figure 14 uses saved 500-epoch California checkpoints and 4,096 fixed noisy training observations. CURB is disabled during measurement. The surfaces span two fixed orthogonal first-hidden-bias directions with unit RMS. The independent quantitative test uses 32 shared Gaussian all-parameter directions per seed, identical absolute displacement norms, and symmetric loss rise $[L(\theta+d)+L(\theta-d)]/2-L(\theta)$. The CURB/ERM mean-rise ratios at norms 0.005, 0.01, 0.03, and 0.1 are 0.0053, 0.0041, 0.0030, and 0.0120. Classification confirmation uses validation-selected Chebyshev/Gaussian checkpoints and unregularized CE. Parameterization and perturbation normalization are fixed within each comparison.

Complementary four-grid regression confirmation. The oscillatory regression experiment uses $[2, 24, 24, 1]$, quadratic splines, a SiLU branch, 1,024 training inputs, 20% corrupted labels, 600

epochs, and fixed CURB strength 1. All 12 paired comparisons across $G \in \{8, 10, 12, 16\}$ and three repetitions favour CURB in clean-test MSE (Table 1). Measured 61×61 loss surfaces for one paired repetition use the same two orthogonal first-hidden-bias directions, unit-RMS direction normalization, and displacement range $[-0.5, 0.5]^2$. All eight surfaces share loss and colour scales; no smoothing or separate vertical rescaling is applied. An independent test with 24 random all-parameter directions per checkpoint gives CURB/ERM mean local second-finite-difference ratios of 0.0133–0.0619 across the 12 pairs.

Computational cost. We measure optimizer-update cost using PyTorch. Each method uses the current canonical implementation and a preloaded batch of 1,024 images, 10 warm-up updates and 20 timed updates, with device synchronization at both timing boundaries. Three repetitions randomize method order and bracket each block with ERM measurements. Costs include perturbation generation, all forward/backward passes and optimizer operations; data loading, checkpointing and evaluation are excluded. For each geometry we report the median relative cost, normalized to the bracketing ERM measurements (ERM $= 1\times$); the main table averages these ratios over seven geometries. CURB uses the fast cell-local implementation in these measurements.

Table 5: **Isolated timings compare optimizer-update cost at matched geometry.** Cost relative to ERM; lower is faster. Entries are medians over three bracketed repetitions.

(G, p)	ERM	Input noise	DropKAN	SAM	ASAM	CURB
$(3, 15)$	$1.00\times$	$0.99\times$	$1.13\times$	$1.97\times$	$2.01\times$	$1.46\times$
$(6, 12)$	$1.00\times$	$0.99\times$	$1.22\times$	$1.99\times$	$1.97\times$	$1.49\times$
$(8, 10)$	$1.00\times$	$0.99\times$	$1.34\times$	$2.02\times$	$1.99\times$	$1.51\times$
$(10, 8)$	$1.00\times$	$0.98\times$	$1.42\times$	$1.92\times$	$2.11\times$	$1.48\times$
$(12, 6)$	$1.00\times$	$1.00\times$	$1.55\times$	$1.92\times$	$2.02\times$	$1.30\times$
$(14, 4)$	$1.00\times$	$0.98\times$	$2.03\times$	$1.96\times$	$2.00\times$	$1.39\times$
$(16, 2)$	$1.00\times$	$1.01\times$	$3.30\times$	$1.99\times$	$2.01\times$	$1.49\times$

Frequency-suite endpoints. The frequency benchmark comprises nine tasks, six grids, and three paired repetitions. Training durations match within clean/noisy pairs and can differ between grids.

D Regression breakdown and learned fields

D.1 Grid-resolved regression results

Table 6: **Grid-resolved comparisons distinguish smooth and oscillatory regression.** Clean-test MSE, three-seed mean $\pm$ SD, 600 epochs. The main table first averages grids within each seed and then reports the mean and SD over seeds.

Task	G	ERM	SAM	ASAM	CURB
Oscillatory	8	$.1423_{\pm.0021}$	$\mathbf{.0436}_{\pm.0022}$	$.0468_{\pm.0033}$	$.0543_{\pm.0040}$
Oscillatory	10	$.1336_{\pm.0039}$	$.0593_{\pm.0030}$	$.0390_{\pm.0019}$	$\mathbf{.0380}_{\pm.0061}$
Oscillatory	12	$.1159_{\pm.0023}$	$.0671_{\pm.0019}$	$.0369_{\pm.0077}$	$\mathbf{.0267}_{\pm.0047}$
Oscillatory	16	$.0896_{\pm.0042}$	$.0715_{\pm.0016}$	$.0249_{\pm.0033}$	$\mathbf{.0194}_{\pm.0037}$
Smooth	16	$.0956_{\pm.0007}$	$.0249_{\pm.0055}$	$\mathbf{.0171}_{\pm.0020}$	$.0253_{\pm.0026}$
Smooth	24	$.0790_{\pm.0012}$	$.0237_{\pm.0038}$	$\mathbf{.0205}_{\pm.0014}$	$.0432_{\pm.0034}$
Smooth	32	$.0747_{\pm.0024}$	$.0217_{\pm.0032}$	$\mathbf{.0213}_{\pm.0019}$	$.0604_{\pm.0031}$
Smooth	48	$.0742_{\pm.0035}$	$\mathbf{.0225}_{\pm.0020}$	$.0229_{\pm.0015}$	$.0663_{\pm.0029}$

D.2 Darcy predictions and spatial errors

The figure shows a fixed test field after 500 epochs. The backbone is identical; SAM radius 0.2 and CURB strength 1 are validation-selected. No spatial smoothing is applied. Table 1 reports the 1,000-field averages.

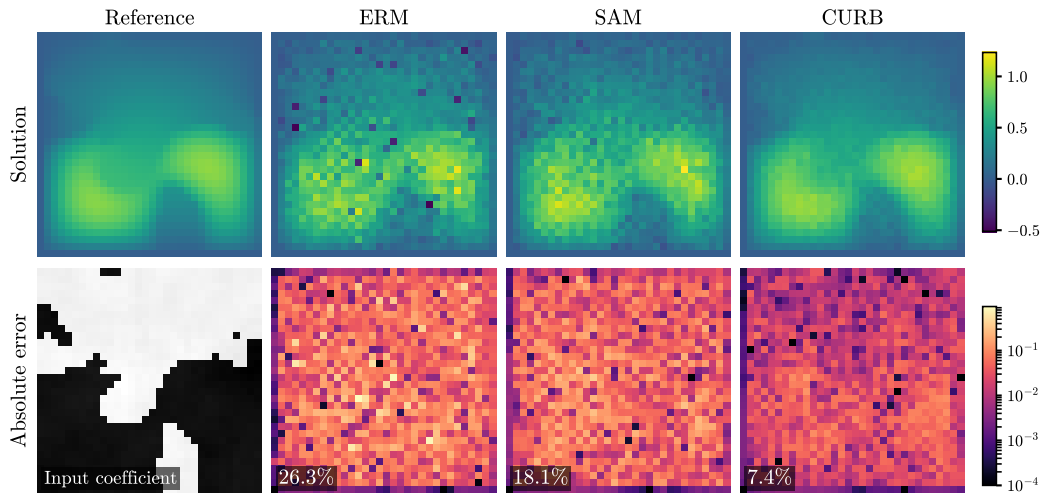

Figure 15: **CURB preserves solution structure and reduces artifacts.** Top: reference and learned Darcy solutions, sharing a colour scale. Bottom: input coefficient and absolute errors (shared log scale). Percentages are this field's relative L^2 errors, not dataset averages.

D.3 Dimensionality and frequency: observation benchmark

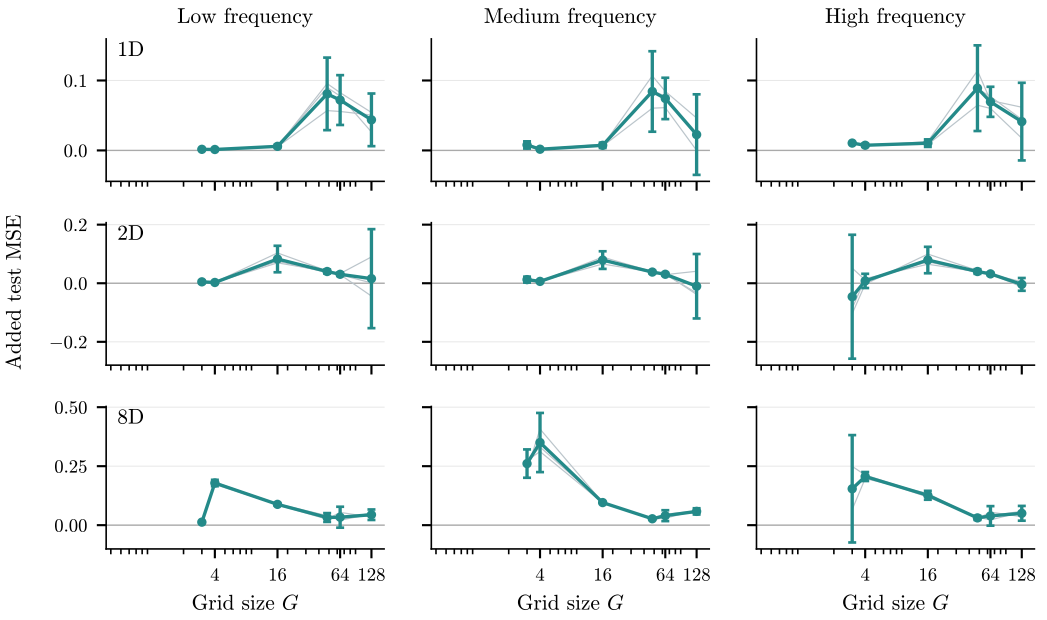

Figure 16: **The memorization problem extends across dimensions and frequencies.** Added MSE is noisy-minus-clean ERM test MSE on manufactured Poisson fields. Light curves: all three paired seeds; dark means: pointwise 95% Student-t intervals. This function-regression benchmark uses ERM throughout and covers all six grid sizes.

E AI usage

Writing: ChatGPT 6 is utilized in trimming the main text to 9 pages, and removing latex errors.

Experiments: ChatGPT 5.6 and 6 is used for running the experiment sweeps.

Theory: ChatGPT 6 is used for generating counterexamples to earlier versions of the theorems, and validating the final versions.