PLOTTABILITY IS NOT IDENTIFIABILITY: AUDITING THE RETRAINING STABILITY OF KAN EXPLANATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Kolmogorov–Arnold Networks (KANs) are promoted as interpretable because every edge is a plottable univariate function. We audit that claim by asking whether the plots survive retraining, and find that on tabular data the stability credited to KANs tracks *additivity*; we detect no contribution from the edge architecture.

A plotted shape is worth reading only if retraining an equally accurate model reproduces it. We measure this retraining stability, a lower bound on disagreement over the Rashomon set, under a leakage-free protocol across 14 tabular datasets, five splits, three shape extractors (PDP, ALE, SHAP) and three bases. Spline and boosted additive models reproduce their plots; a deep KAN's do not, on 9/9 regression and 5/5 classification datasets, the gap clearing ± 2 SE on 13/14 and never inverting at a common held-out loss radius. A deep KAN is no more stable than a deep MLP (7/14); an apparent 20% KAN edge at high capacity coincides with the MLP carrying more parameters and is not detected near parity, against an additivity effect of 1.7–$33.8\times$. No deep model of B-spline, RBF or Chebyshev basis approaches the spline additive reference. A matmul-only profiler misses 53–81% of a deep KAN's operations; counting them, its arithmetic is 0.8–$2.4\times$ the MLP's.

We then draw the boundary of our own standard: the hierarchy inverts where an additive model is misspecified, the accuracy gap is often not about interactions, and controlling knot placement leaves the direction intact on 14/14 while halving a magnitude we decline to claim. The ordering holds from 6k to 50k rows.

Finally we turn the standard on itself: four of five interaction screens reproduce their own top pairs across seeds (median Jaccard 1.00, the fifth failing it) while agreeing with one another at 0.20, and KAN's own prune-and-symbolify workflow—on Feynman equations, where a formula exists to find—does not converge on a structure across seeds and recovers the generating equation on at most one seed in five, while every seed's formula predicts alike. Cross-seed stability is necessary but not sufficient; a method can be perfectly reproducible and still arbitrary.

1 INTRODUCTION

Kolmogorov–Arnold Networks (Liu et al., 2025) replace the fixed node activations and linear edge weights of an MLP with learnable univariate functions on edges, summed at nodes. A central selling point is interpretability: each learned edge function can be plotted and read. Yet "plot the edges" is meaningful only if the plot is *a property of the data and the model class*, not of the random seed. Decades of work on generalized additive models (GAMs) warns that equally accurate additive fits from different algorithms can learn contradictory shape functions (Chang et al., 2021), an identifiability problem that no amount of plotting resolves. KAN is the test case because it makes the plottable-therefore-interpretable claim most explicitly; the finding is about the claim.

The stakes are concrete. Fig. 1 shows a deep KAN whose partial-dependence plot for one feature *reverses direction* at high values between two retrainings on the same data, while the additive model's plot is identical. It is an *existence* demonstration (the caption states how it was selected), but a shape that can flip its message with the seed is not one to build decisions on.

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.

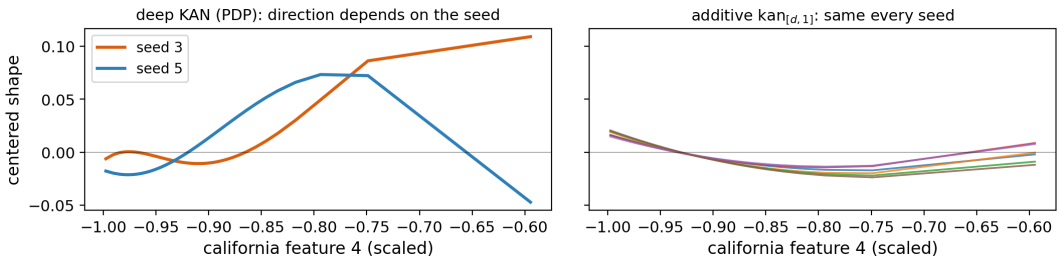

Figure 1: Why identifiability matters (California Housing; an *existence* demonstration). For the feature with the most negative cross-seed shape correlation, the two most-opposed of six deep-KAN retrainings disagree on the *direction* of the effect at high values (left); the additive $\text{kan}_{[d,1]}$ gives the same shape every retraining (right). A separate six-seed run, `experiments/harm_figure.py`.

We start from a structural observation that is folklore but rarely operationalized: a width-$[d, 1]$ KAN, one learnable univariate function per input edge summed at a single output, *is* a GAM, $g(y) = \beta_0 + \sum_j \phi_j(x_j)$, with B-spline shape functions—a Neural Additive Model (Agarwal et al., 2021) with splines for the per-feature MLPs. The KAN's claimed interpretability is the GAM's, and the GAM's identifiability question applies directly. A hidden layer breaks the equivalence: it buys interactions, but the per-edge plots no longer describe the function additively.

This paper treats "interpretable" as a claim with a test and reports what it returns. KAN's interpretability is claimed at four levels—edge functions, their marginal summaries, the interactions a model is said to have discovered, and the expressions extracted from it—and we put all four to one test: does an equally accurate retraining reproduce the object? At none of the four is the answer the one the KAN literature implies, and at none is the property KAN's. On the tabular designs we study, what buys a stable plot is additivity.

Contributions.

- **A falsifiable standard** (§3): a cross-seed shape-disagreement metric over predictive-equivalent retrainings, a leakage-free protocol, and an API that scores any glass-box class with a shape extractor.
- **The hierarchy, measured** (§4): 9/9 regression and 5/5 classification, on five splits, under three extractors and at every non-additive capacity tried; 11/14 under a scale-invariant metric, reported as the claim's strength.
- **Additivity, not the edge architecture** (§5): no consistent deep-KAN advantage over a deep MLP is detected (7/14, and in a capacity-sensitivity sweep) against an additivity effect of 1.7–33.8×; no deep model of three bases approaches the spline additive reference; and 53–81% of a deep KAN's operations are invisible to a matmul-only profiler.
- **The standard applied to itself** (§7, §8): interaction screens reproduce their own answer yet agree with each other at 0.20, and KAN's prune-and-symbolify workflow does not converge on a structure across seeds while its formulas predict alike: cross-seed identifiability is necessary but not sufficient.
- **Where it does not apply** (§8, §9): the hierarchy inverts where the additive model is misspecified, the accuracy gap is often not about interactions, and our own additive-first model is Pareto non-dominated on only 3/14.

2 Background and Related Work

KAN as a learnable GAM. KANs place learnable univariate functions on edges (Liu et al., 2024; 2025); at width $[d, 1]$ this is precisely a spline GAM (Stone, 1985), and the interpretability claim is inherited from that literature rather than new to KANs.

Fair comparison. Yu et al. (2024) show that KAN's accuracy advantage over MLPs largely disappears once parameters and compute are matched, and that what remains is the B-spline basis rather than the edge architecture; we extend that verdict to interpretability.

Identifiability of additive models. Concurvity—approximate dependence among features, the additive analogue of collinearity—makes additive components non-unique (Buja et al., 1989); the classical remedy is a non-degeneracy condition on the design distribution (Stone, 1985), and this gap is what pins a shape.

Multiplicity of equally-good models. The Rashomon set (Fisher et al., 2019) of near-optimal models can support contradictory explanations (Marx et al., 2020; D'Amour et al., 2022; Rudin et al., 2024); identifiability is how much a plotted shape varies over that set, hence a metric over retrainings rather than one fit. Closest to us, Cavus et al. (2025) aggregate the PDPs of an AutoML Rashomon set into a band and find that on most of 35 regression tasks it covers under 70% of the best model's PDP; we hold the model class and training procedure fixed, vary only the seed, and compare classes. Post-hoc explanations are separately fragile to input or explainer perturbations (Ghorbani et al., 2019; Alvarez-Melis & Jaakkola, 2018); we perturb neither.

Interactions in additive models. GA2M/EBM (Lou et al., 2013; Nori et al., 2019) and NodeGAM (Chang et al., 2022) add pairwise terms to a GAM; EBM fits main effects to convergence before adding pairs, which decides our frontier comparison (§9). Interaction terms are non-unique without a convention; functional ANOVA (Hooker, 2007) and purification (Lengerich et al., 2020) fix one after the fact, and we impose it during training.

Symbolic regression. Dedicated symbolic regression already exposes many expressions per dataset as a Pareto front *within one run* (Udrescu & Tegmark, 2020; Udrescu et al., 2020; Cranmer, 2023); ours is a multiplicity across *retrainings*: the expression one run settles on is not another's. A recent variant that gates each KAN edge to one primitive reports improved "structural identifiability" from single runs (Faroughi et al., 2026); whether its primitives agree across retrainings is the question §8 asks, and there it is untested.

KANs read as GAMs. Polson & Sokolov (2025) argue for an additive form too, though additive over a fixed embedding rather than over the raw inputs x_j that a shape plot is drawn against; their claim is architectural, ours is measured: granting additivity, is the plot reproducible? Clinical work has reached the same prescription: Almodóvar et al. (2025) read KAN shape plots as clinical evidence and, for that purpose, propose an explicitly additive KAN, the class our measurements single out.

What is known, and what this paper adds. That additive models can be identifiable and non-additive ones need not be is folklore; that equally accurate GAMs can disagree (Chang et al., 2021; Lengerich et al., 2020) and that Rashomon-set explanations vary (Fisher et al., 2019; Rudin et al., 2024) is established. What has not been done is to measure the distinction as retraining stability, under a leakage-free protocol, across model families and bases at matched accuracy, with KANs as the test case; to separate the additivity effect from the architecture (§5); to test, and refute, the reading that the result is definitional (§4); and to apply the same test to discovered interactions and extracted formulas, where cross-seed stability is necessary but not sufficient (§7, §8)—an observation about interpretability audits in general rather than about KANs.

3 Identifiability: The Standard and How We Measure It

Definition. Let $\mathcal{R}_\epsilon$ be the Rashomon set of models within ϵ of the best achievable risk. For a shape extractor S_j returning feature j's plotted curve, define its *disagreement* $D_j(\epsilon) = \sup_{f,f' \in \mathcal{R}_\epsilon} \|S_j(f) - S_j(f')\|$ after centering; a plot is identifiable when D_j is small. What we can measure is narrower: $\hat{D}_j$, the mean pairwise disagreement over retrainings of one training procedure on fixed data, which lower-bounds D_j when every retraining lands in $\mathcal{R}_\epsilon$. A large $\hat{D}_j$ therefore proves non-identifiability; a small one shows retraining stability and only bounds D_j from below.

Every "identifiable" in the results is a claim about $\hat{D}_j$, and the additive/deep *ordering* is an ordering of $\hat{D}_j$; we do not claim the supremum is small for any class.

Proposition 1 (App. A) names the two forces: an additive model's shapes are pinned by the design's concurvity gap γ, while at a fixed risk radius a non-additive model's PDP gap can grow as $1/(1-\rho)$ with the correlation ρ and its ALE gap stays bounded (part iii; part ii is the exact-degeneracy limit).

Protocol. No reported number trains, early-stops or scales on the held-out test split: features are min-max scaled on train only, early stopping uses a validation split, interaction screening sees only the train residual. Two things do read the test metric and are named where used: the seed-pair filters of §4, which select *which retrainings* enter a control, and the label-free classification normalizer of §8. Datasets are capped at 12k rows, non-numeric columns dropped (Table 8). For each dataset and model we retrain over seeds and report the mean pairwise cross-seed disagreement on a fixed quantile grid, centered and normalized by the target scale so that model families are comparable, under partial dependence and under accumulated local effects (ALE, Apley & Zhu, 2020), which is less biased under correlated inputs.

Why partial dependence, and not the edge plots. KAN's claim is about the learned per-edge functions $\phi_{l,i,j}$, but in a deep KAN the edges are not identifiable even in principle (permuting hidden units, or rescaling layer-0's edges against layer-1's, leaves the function untouched), so comparing them across seeds is not well-posed. Measured, they agree far less than the partial dependence does even after an oracle matching of hidden units (App. B); partial dependence is the *conservative* reading.

Metric. Our headline nRMSE is amplitude-sensitive; we also report a scale-invariant Pearson variant, and both counts, not the flattering one. Intervals on nRMSE cover seed-to-seed variation only, and the data-draw term is larger: an independent subsample of the same dataset moves the additive model's value by $3\times$ (`house16` at n=11k against 12k, 0.0033 against 0.0099), one reason we claim the ordering and not the magnitude. The sparsified and `pykan` paths of §8 add a third term: a five-seed dispersion statistic there moves by up to 17% between identical runs (Table 9).

Configuration, and what "deep" means here. The deep KAN is $[d, 16, 1]$: one hidden layer of width 16, `grid_size`$= 5$, spline order 3, a fixed uniform grid on $[-1, 1]$, and `update_grid` never called (`pykan`'s knot refinement is a control in §8); the additive model $[d, 1]$ uses the same basis. These are small models ($1{,}440$ and 80 parameters on California; the sweep reaches 57k), and the objection that larger ones would be identifiable runs against the sweep's own direction (§5). "Deep" is shorthand for *non-additive* rather than for depth as such: across six non-additive configurations up to $[d, 64, 32]$ and $[d, 256, 128, 64]$, the most stable is still $1.7\times$ the additive model's disagreement at minimum (Table 3). The primary runs use no sparsification penalty (§8 tests KAN's own sparsify-and-prune workflow).

The shape extractors are verified exact *given their background sample*; re-drawing it moves a deep model's nRMSE by up to 20% (`kin8nm`), well inside the additive/deep gap but not negligible for any single magnitude.

4 The Hierarchy, Measured

Under the leakage-free protocol the additive$<$deep identifiability hierarchy holds on **9/9 regression and 5/5 classification** datasets for the spline GAM (Tables 1, 2; Figs. 3, 4). Additivity does not guarantee it: scored the same way, EBM beats both deep models on 13/14 but *NAM* only on 9/14, losing on `cpu_act`, `kin8nm`, `pumadyn`, `house16` and `spambase`. A per-feature MLP is an additive model whose own shapes are not pinned; on these tabular designs the spline and boosted additive models were the stable ones, and that is the finding, not a necessity claim (§8 shows the inverse case). That comparison also cuts at our title: the spline GAM beats NAM on **14/14**, so within the additive class the shape function's own form still matters. The architectural claim we test is the narrower one: a deep KAN is no more identifiable than a deep MLP, at the primary configurations or at parameter parity (§5). For the spline GAM a one-sided binomial sign test gives p=2.0×10^{-3} regression, p=3.1×10^{-2} classification, p=6×10^{-5} combined. The suite is five of ten registry

candidates; the other five hold on 4/5 under the same protocol (Table 14). It reproduces on all five independent train/val/test splits for every dataset (Tables 21, 22). Read by one object for every class, the PDP of the full predictor with additive and GrowKAN models included, it holds on 14/14; for classification it also holds on the probability scale (5/5) and with a train-rows-only normalizer (5/5; Table 15). For a reader the gap is mostly magnitude and shape, not sign: where both retrainings give a material interquartile effect, a deep KAN's readings disagree in sign in 1.0% of cases (9/938, on 2 datasets), a deep MLP's in 0.7%, the additive model's never (0/946); reversals like Fig. 1 are rare.

The result is not an artifact of the extractor: under ALE the gap survives on every dataset (Table 23; ALE nRMSE is usually below PDP's but far above the additive model's), and under SHAP dependence curves it holds on 14/14 (Table 10). Nor is it an artifact of comparing unequally-accurate retrainings, and this needs more than the usual check. Restricting to *predictive-equivalent* seed pairs, both within 10% of the model's best held-out metric, leaves 9/9 and 5/5 under both extractors (Table 24); but that filter binds on only `airfoil`, `phoneme` and `sylvine`, where the deep KAN's seed spread reaches 13–23% of its best, and on the other eleven (spread under 8%) it removes nothing.

The two classes do not span the same Rashomon set, and that is the objection to answer. The deep KAN's held-out loss ranges 3.8–$24.3\times$ wider across seeds than the additive model's, on 14/14 datasets. Since Proposition 1(i) bounds an additive model by $C\sqrt{\epsilon/\gamma}$, a sceptic can read our whole effect as optimization variance, and the arithmetic is on their side: dividing each measured ratio by $\sqrt{\epsilon_{\text{deep}}/\epsilon_{\text{add}}}$ gives a median of 1.00. So we hold both classes to a *common* radius: with $\epsilon^\star$ the additive model's own held-out loss spread, keep in each class the seeds within $\epsilon^\star$ of that class's best seed (a held-out proxy for the distance from the class optimum that the proposition's ϵ is). The direction never inverts (8/8, 11/11, 12/12 at $\epsilon^\star$, $2\epsilon^\star$, $3\epsilon^\star$; ten seeds, Table 13), but it is underpowered: at $\epsilon^\star$ only 8 of 14 datasets keep a deep pair, five of those eight on a single one, and the other six carry no evidence either way. Equally good deep models, matched to the additive model's own tolerance, still disagree about the shapes, on as much evidence as that tolerance leaves.

The ordering is not a bare sign count. Errors on the cross-seed nRMSE come from a leave-one-*seed*-out jackknife, since its 10 pairwise terms come from 5 fits. The additive/deep gap clears ±1 SE on **14/14** datasets and ±2 SE on **13/14** (`sylvine` is the exception; at ten seeds the direction holds on 14/14 and `sylvine` clears, while the ratios move up to 57%; Table 7).

The result is not definitional. One can object that this was settled in advance: a non-additive function has no well-defined one-dimensional summary, so of course a marginal read of one varies. That objection predicts something testable—instability should be worst where the accuracy gap is genuinely about interactions—and the zero-interaction control of §8 measures that fraction per dataset. The prediction fails with its sign reversed: the control's closed-fraction correlates with the deep/additive nRMSE ratio at Spearman $\rho{=}+0.92$ ($p{<}10^{-3}$, $n{=}10$; leave-one-out range $[+0.88, +0.95]$). The deep model's plots are *least* reproducible where its advantage needs no interactions at all: California closes 99% of its gap without them yet is the least identifiable ($24.5\times$), while `kin8nm`, `pumadyn` and `airfoil`, whose gaps *are* interaction-driven, show the mildest ratios (1.9–$5.0\times$). The controlled version is Fig. 2 (left): with the data held fixed, the deep PDP's instability moves with concurvity, not with interaction energy. That is a synthetic result we report as a mechanism *candidate*, since across the fourteen real designs a linear concurvity proxy does not predict the ratio (Spearman -0.09, $+0.39$; §10).

Under the scale-invariant Pearson variant the count is **11/14** rather than 14/14 ($8/9$ regression, $3/5$ classification); the direction is unchanged and the regression result remains significant, the classification half does not: both classes correlate highly across seeds (median 0.96 against 0.995), and nRMSE separates them by the vertical spread a plot reader sees. We report the smaller number as the claim's strength.

5 Additivity, Not a Detected Architectural Effect

If deep KANs were interpretable *as KANs*, a deep KAN should be more identifiable than a deep MLP. At the primary configurations ($[d, 16, 1]$ against $[d, 64, 64, 1]$) it is not: the deep KAN wins on

Table 1: **Leakage-free regression results** (held-out test RMSE, lower is better; one fixed split per dataset, seeds vary model init). The additive<deep hierarchy holds on 9/9; the gap clears ±1 SE on 9/9 and ±2 SE on 9/9. GrowKAN k=2 uses train-only-detected pairs. Errors: seed SE for accuracy, leave-one-seed-out jackknife for nRMSE (a dispersion statistic). Across seeds the additive model's accuracy spans at most 0.58% of its mean and the deep KAN's 17.4%: the additive model's held-out metric barely moves across initializations. GrowKAN's nRMSE is a point estimate: the sweep logs its mean cross-seed disagreement but not the pairwise values a jackknife needs.

	test RMSE			mains nRMSE			
dataset	GAM	deepKAN	GrowKAN	GAM	GrowKAN	deepKAN	deepMLP
california	0.654±0.001	0.534±0.004	0.592±0.005	0.00574±0.00115	0.0255	0.141±0.016	0.0975±0.0152
airfoil	4.93±0.00	1.93±0.06	2.74±0.02	0.0183±0.0054	0.0440	0.0908±0.0133	0.0705±0.0088
wine	0.741±0.000	0.718±0.003	0.736±0.000	0.00745±0.00190	0.00994	0.0509±0.0173	0.0222±0.0019
cpu_act	2.93±0.00	2.42±0.01	2.59±0.00	0.00147±0.00020	0.00619	0.00915±0.00260	0.00555±0.00073
kin8nm	0.194±0.000	0.0732±0.0005	0.180±0.000	0.00356±0.00035	0.0206	0.00933±0.00078	0.00940±0.00098
elevators	0.00238±0.00000	0.00189±0.00000	0.00225±0.00000	0.0105±0.0012	0.0210	0.0380±0.0093	0.0439±0.0071
pumadyn	0.0340±0.0000	0.0221±0.0000	0.0214±0.0000	0.00362±0.00038	0.00748	0.00697±0.00036	0.0114±0.0009
wind	3.02±0.00	2.98±0.01	3.01±0.01	0.00904±0.00078	0.0275	0.0242±0.0041	0.0192±0.0034
house16	40444±14	37618±302	39962±149	0.00835±0.00087	0.0152	0.0407±0.0074	0.0367±0.0038

Table 2: **Leakage-free classification results** (held-out test 1-AUC, lower is better; one fixed split per dataset, seeds vary model init). The additive<deep hierarchy holds on 5/5; the gap clears ±1 SE on 5/5 and ±2 SE on 4/5. GrowKAN k=2 uses train-only-detected pairs. Errors: seed SE for accuracy, leave-one-seed-out jackknife for nRMSE (a dispersion statistic). Across seeds the additive model's accuracy spans at most 4.01% of its mean and the deep KAN's 20.6%: the additive model's held-out metric barely moves across initializations. GrowKAN's nRMSE is a point estimate: the sweep logs its mean cross-seed disagreement but not the pairwise values a jackknife needs.

	test 1-AUC			mains nRMSE			
dataset	GAM	deepKAN	GrowKAN	GAM	GrowKAN	deepKAN	deepMLP
phoneme	0.132±0.000	0.0772±0.0017	0.100±0.001	0.0122±0.0061	0.0559	0.0698±0.0113	0.0662±0.0192
magic	0.110±0.000	0.0786±0.0005	0.0934±0.0003	0.0131±0.0025	0.0175	0.0473±0.0067	0.0539±0.0070
jasmine	0.195±0.000	0.206±0.001	0.187±0.001	0.00313±0.00041	0.0181	0.0176±0.0024	0.0253±0.0046
sylvine	0.0189±0.0002	0.0188±0.0007	0.0177±0.0001	0.0121±0.0018	0.0159	0.0248±0.0047	0.0347±0.0051
spambase	0.0243±0.0000	0.0237±0.0003	0.0241±0.0001	0.00191±0.00028	0.00224	0.00442±0.00040	0.00631±0.00089

only **7/14** datasets (two-sided sign test p=1.0; median MLP/KAN cross-seed nRMSE ratio 0.98; hierarchy tests are one-sided because their direction was fixed in advance, every KAN-versus-MLP test is two-sided). Replacing the B-spline basis with RBF (Li, 2024) or Chebyshev polynomials (SS et al., 2024) leaves the ordering unchanged (Table 27; the non-additive RBF and Chebyshev models are basis-embedded MLPs, so structure changes with the basis), and no deep model of any basis at widths 16–256 comes within $1.7\times$ of the spline additive reference (Table 11). The basis does matter for the *additive* model: Chebyshev shapes are 1.1–$7.5\times$ less reproducible than spline shapes, enough that within that family the ratio reaches 1 at width ≥ 64 on two of six datasets: a global polynomial basis forfeits part of the additive advantage, Proposition 1(i)'s γ being basis-dependent. In the settings we evaluated, the spline additive model was the most stable reference. Identifiability is thus a property of the shape-function family, additivity one axis and the basis another: on `spambase` a deep KAN's PDP (0.0044) is more identifiable than either EBM's (0.0072) or NAM's (0.0136) main effects, and only the spline GAM (0.0019) beats it. The NAM comparison carries a caveat we cannot remove: at its default width it has $224\times$ the additive KAN's parameters on every dataset, and capacity is a sufficient explanation for its looser shapes.

Capacity complicates this. Sweeping both architectures from their smallest to largest configuration (Table 3), the deep MLP's partial dependence gets *less* reproducible with size (worse on 11/14, median $1.30\times$) while the deep KAN's does not (median $0.97\times$), so at the largest configurations the KAN is the more stable of the two on 10/14 (median ratio 0.82; Wilcoxon on the log-ratio p=0.03), and the same shift appears after validation-selected tuning (12/14, p=0.013; Table 4). That reads as a 20% architectural effect, but neither comparison is parameter-controlled, the rule we credit Yu et al. (2024) with: the MLP carries $1.55\times$ the KAN's parameters at the largest configurations and $1.68\times$ after tuning. At the swept pairing whose *median* parameter ratio is nearest one ($[32]$ against $[64, 64]$, $1.19\times$; per dataset it ranges 0.43–$2.40\times$, so this is not per-task control) the KAN leads on $9/14$, median ratio 1.04, bootstrap CI $[0.92, 1.23]$. We had no pre-registered equivalence margin, so we do not claim the two are equal, and this is a capacity-sensitivity analysis, not a capacity control:

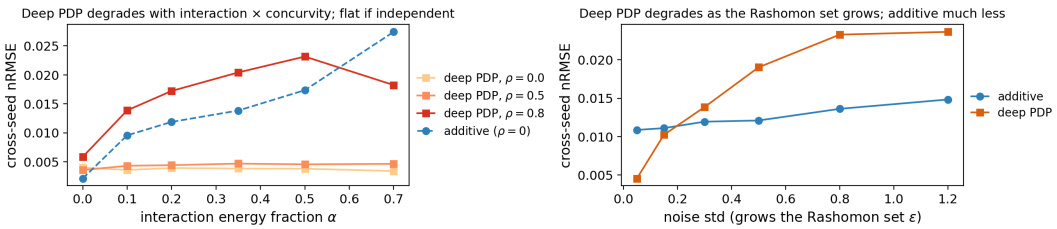

Figure 2: A mechanism candidate, on controlled synthetic data. **Left:** the deep PDP's cross-seed disagreement rises with interaction energy α only under strong concurvity (ρ=0.8); at $\rho \leq 0.5$ it moves at most $1.3\times$, while the *additive* model degrades with the interaction it must absorb (blue). **Right:** as label noise widens the retrained-loss spread, the deep PDP degrades $5.2\times$; the additive shape rises $1.4\times$. One data realization per cell, 8 init seeds.

no consistent architectural advantage was detected in this setting, and 1.23 bounds the median ratio over these 14 tasks, not any one task's, against an additivity effect of 1.7–33.8× (median 3.2×; Table 3).

The cost argument does not hold either. KAN's efficiency claim rests on parameter counts, but a profiler that counts only the matmul family skips the de Boor recursion, where a KAN spends its operations: 1296 matmul operations per sample against a full count of 6680, so 81% **of a deep KAN's work is invisible** here, and 53–81% across both profiled datasets and conventions (App. E). Whether a multiply–accumulate counts as one operation or two flatters one architecture or the other, so we give both (Table 17) and claim only what survives: a deep KAN's arithmetic is 0.8–2.4× a deep MLP's. Operation counts are not throughput, and ours are too noisy to time; what is shown is that parameter counts and matmul-only profilers cannot decide the efficiency question, not its answer.

6 Mechanism

Proposition 1 identifies two forces, and controlled synthetic sweeps separate them (Fig. 2). Concurvity unpins a deep model's marginals: sweeping interaction energy α, the deep PDP moves at most $1.3\times$ at $\rho \leq 0.5$ but up to $4.0\times$ at ρ=0.8 ($3.1\times$ at the largest α); interaction energy costs the *additive* model instead (nRMSE 0.002→0.027). Label noise widens the spread of retrained losses and acts on the deep model alone: at fixed interaction and concurvity it degrades the deep PDP $5.2\times$ and the additive shape $1.4\times$. That is a finite-sample effect, not the proposition's ϵ: at the population level additive noise cancels from $L(f) - \inf L$.

More data shrinks the spread of retrained losses, and the contrast shows in how shapes stabilize with n. On `house16` from 6k to 22k rows the additive shape's disagreement falls as $n^{-1.34}$ while the deep PDP's *rises* ($n^{+0.27}$), taking the ratio from $1.7\times$ to $14.1\times$; on `magic`, $3.2\times$ to $16\times$. More data makes a deep KAN's plots *less* reproducible here, though not on `fried` or `elevators`, where both improve. The seed spread of held-out loss is consistent with this (App. A): the additive model's shrinks as $n^{-1.61}$ on `house16`, the deep one's as $n^{-0.17}$. That is the learner's spread, not the class's Rashomon set.

7 The Same Test, One Level Up: Which Interactions Are Found

The growth path also diagnoses interaction *order*, and reads `kin8nm` and `elevators` as higher-order (Tables 20, 35); we state that verdict only for those two, because California and `pol` flip with the *screen*, so an order verdict must carry its screening scope.

Which interactions are found is not identifiable either—and cross-seed stability does not detect it. We apply our own standard to the discovered pair set, using five screens on all 14 datasets: four share the residual-binning *rule* and differ only in whose residual they read (our additive base, boosted trees, a linear model, a small MLP); the fifth is EBM-GA2M's internal criterion. We measure each

screen refit across seeds, different screens on the same data, and the chance overlap of two uniform draws from $\binom{d}{2}$ (Table 16).

Median within-screen agreement is 1.00 and median between-screen agreement is 0.20, with 44 of 140 method pairs at or below chance. Two of the five are deterministic given the data—the linear base has no randomness, and the boosted one resamples only above 10k rows—so their 1.00 is arithmetic, not evidence. The attribution is the surprise: the rule shows no separate effect here, since screens sharing it agree at 0.20, as do screens that do not (a comparison, not an experiment that isolates the rule). *Whose residual you look at* determines what you find; without ground truth we cannot say which screen is wrong, only that the set is a property of the base model as much as of the data. Two qualifications: the effect grows with dimension, and the MLP-residual screen fails our own test (within-seed agreement 0.69, 0.00 on `elevators`).

This qualifies the standard this paper proposes: **cross-seed identifiability is necessary but not sufficient**. A screen can score 1.00 against itself and still report an artifact of its base model, and no retraining reveals it. For shape plots the three extractors agree on the *ordering*; whether they agree on the curves themselves we have not measured, and we report our own screen as an open problem.

8 Where Identifiability Is Not a Virtue

The hierarchy inverts where the additive model is misspecified. On the Feynman equations—closed-form physics with independent variables and genuine multiplicative structure—the additive model is the unstable one on all $10/10$ (Table 25): with independent features the deep model's marginals are well-defined while the additive fit must absorb interaction it cannot represent (whether the instability is that misspecification or the optimizer's response to it, we cannot separate). Identifiability is a virtue only where the additive class is adequate; elsewhere an identifiable plot is a confidently wrong one.

The gap is often not about interactions. A zero-interaction control (an EBM restricted to main effects) closes a median 26% of the accuracy gap, from $\approx 0\%$ on kin8nm, elevators and pumadyn to 99% on California, where the deep model's advantage is additive-fitting capacity rather than interactions.

The direction is robust; the magnitude is a configuration choice. A fixed uniform knot grid on min-max-scaled inputs leaves skewed features with most knots in empty regions (96% of features on `spambase`). Refitting *both* models on rank-transformed inputs, which spreads the data across knots, the hierarchy still holds on **14/14** (Table 18); the direction is not a knot artifact.

The magnitude is. The deep-to-additive ratio shrinks on 11/14 datasets and its median falls from $4.2\times$ to $2.0\times$; on `wind` it *rises*, $2.7\rightarrow 9.3\times$, because the rank transform makes that dataset's additive shapes four times more reproducible. Jackknife intervals first suggested more than is true: at five seeds the *ratio's* own interval—a stricter test than the overlap of §4—clears 1 by 2 SE on 12/14 under min-max but 7/14 under the rank transform, which at ten seeds is **13/14** under both: a seed budget rather than the control. Letting the KAN refine its knots during training, as `pykan` does, says the same: accuracy improves on most datasets, training breaks on two (`elevators` and `spambase`), and the rest hold 12/12 at median $2.0\times$ (Table 12). The ordering holds on every dataset under every transform and budget, and on every dataset where knot refinement trains at all; the ratio's *value* does not, halving under the rank control and moving 57% with seed budget alone. We claim the ordering and treat any single "$N\times$" figure, ours included, as a statement about a configuration.

KAN's own workflow, under the same test. The interpretability claim is also attached to a workflow—sparsify, prune, fit each surviving edge with a symbolic primitive, read off a formula—so we ran `pykan`'s recipe once per seed (Table 5; App. D). Sparsification alone does not close the gap (California nRMSE $0.141\rightarrow 0.050$ against additive 0.006; Table 19). On real data the full recipe returns a finite formula on only **8 of 15** seeds (8–12 of 15 across three runs; Table 9); no two agree on the formula or on its structure with constants stripped, the per-edge primitive agrees on 9–17%, while their held-out predictions correlate at 0.91–0.97. On four Feynman equations, where a formula exists, 17 of 20 seeds return one; on three equations no two seeds share a structure, and none of the 17 simplifies to the generating equation (distance 0.0098 to a median of 0.10) while predictions

correlate at 0.99–1.00. PySR (Cranmer, 2023) on the same rows is seed-dependent in the same way yet recovers $x_1(x_2-x_3)x_4$ on 2 of 5 seeds, which KAN's workflow reaches (1 of 5) only at four times its tutorial budget. Seed-dependence belongs to searching for formulas in data. We read this narrowly: four equations, fixed complexity weight, two budgets; a different skeleton need not be a different law (no canonical form is imposed), and recovery is judged at a numerical tolerance.

9 What It Costs to Pass the Audit

We built for the standard: an additive-first model that screens interactions on the train residual and penalizes them for absorbing main-effect structure (App. C). It recovers 11–106% of the deep model's accuracy advantage while staying more identifiable than a deep KAN on 10/14, and is Pareto non-dominated on only **3/14** against the full candidate set (Table 34), $4/14$ if EBM is read by PDP rather than its native terms, and 4/9 on regression with every model read by full-predictor PDP (Table 15). The defeat is structural: EBM fits main effects to convergence *before* adding pairwise terms, so its native main effects are bit-identical to its pure-GAM ones at better accuracy—a scheduling decision buys what our penalty pays for.

10 Limitations

Inference discipline. The headline count is metric-dependent: 14/14 under amplitude-sensitive nRMSE becomes 11/14 under the scale-invariant variant, whose classification half ($3/5$) is not significant (§4), and a reader who prefers that reading should take 11/14 as the claim. We report two metrics, three extractors, two tasks and several controls without correction; the primary endpoint—the additive<deep sign test on nRMSE under PDP—is fixed in advance and the rest are robustness checks on it, and its raw $p{=}6 \times 10^{-5}$ stays below $0.05/m$ for any $m \leq 800$, though no correction answers the adaptivity of a design revised in review. Sign tests are across datasets because at five seeds a per-dataset Wilcoxon bottoms out at $p{=}1/32$; they treat our 14 datasets as the sample when they are a design choice, and the 70 dataset×split cells reuse the same raw data, so resampling datasets *and* splits hierarchically gives a median deep/additive ratio of $4.01\times$, CI $[2.62, 5.72]$, with the direction holding on every cell.

What the controls cost. The radius-anchored Rashomon control (§4) never inverts but is underpowered: at $\epsilon^\star$ only 8 of 14 datasets keep a deep pair, five on a single one, and both its radius and its anchor are held-out-error quantities; anchored on validation (five seeds) it keeps a deep pair on four datasets, all holding. The extractors are exact only given a background sample; re-drawing it moves a deep model's nRMSE by up to 20%. The concurvity mechanism of §6 rests on synthetic sweeps of one data draw per cell and does not appear across the real designs (Spearman -0.09, $+0.39$); it is a candidate. The seed intervals omit the data-draw term, which is larger (§3). Each bounds a magnitude; none touches the ordering.

Scope. Classification ratios use a per-class logit normalizer over all inputs; it does not decide the ordering (5/5 on the probability scale and train-only, Table 15). The symbolic workflow is not reproducible run to run even at fixed seeds, so its counts are ranges over three runs (Table 9). Controls beyond the primary sweep are single-split; screening uses 2-D/3-D residual binning, so high-order interactions may be missed. At n=6k–50k over ten seeds the ordering holds on every cell and the ratio grows with n except on `diamonds` (Table 6); beyond 50k rows, 57 features, categorical inputs and networks of this size we have not looked. Everything but the Feynman tests is tabular; PDE settings, `pykan`'s LBFGS training and a wider tuning grid (EBM-GA2M runs at its defaults) are left to future work.

11 Conclusion

Treated as retraining stability, the interpretability credited to deep KANs does not reproduce at any level we measured, from edge plots to extracted formulas, and not only for KANs; whether one fit's plot is faithful *to that fit* we did not test. The practice we ask for is small: report a shape's cross-seed disagreement beside the model's accuracy, and treat one retraining's plot as one draw. We would like the axis adopted, not a model.

ETHICS STATEMENT

This work audits an interpretability claim; it introduces no new data collection and no human-subject research. Every dataset is public and already de-identified: the sklearn California Housing set and OpenML tabular tasks, used as distributed and cited with their identifiers in Table 8. No personal or sensitive attributes are analysed, and no model trained here is proposed for deployment.

The ethical content of the paper is in its finding. Shape plots read off a trained model are already used as evidence in consequential settings—clinical classification among them (Almodóvar et al., 2025)—and we show that for non-additive models those plots are not reproducible across retrainings of equal accuracy, with a feature's effect able to reverse direction between two fits on the same data (Fig. 1). Presenting such a plot as an explanation can mislead a decision-maker who has no way to see the alternatives. Our recommendation is correspondingly conservative: report cross-seed disagreement next to accuracy, and where the plot is the object being read, prefer the most stable model among those whose accuracy meets the task's tolerance. We also report the limits of our own standard, including that it does not detect an interaction screen's dependence on its base model (§7), so that it is not adopted as a stronger guarantee than it is.

Use of generative AI in preparing this work is disclosed separately in the AI use statement below, as ICLR requires.

REPRODUCIBILITY STATEMENT

Every primary (leakage-free) result row records the git commit, whether the tree was clean, the command, package versions, the train/val/test split hash and a content hash of the input data; `reproduce.sh` regenerates every result file, figure and table from a clean checkout, and a pinned `requirements.txt` fixes the environment. The identifiability protocol is released as an API (`kanlab.benchmark.identifiability_score`) that scores any torch glass-box model class with a shape extractor. Three checks are asserted by the test suite: every hand-written number in the prose is re-derived from the result files at every place it appears; every figure the paper includes is written by some `reproduce.sh` step; and every result file the body cites carries a clean-tree provenance stamp. One preprocessing constant was set from a test observation and should be named: scaled features are clipped at ± 10 because a near-constant train feature with a test outlier once diverged (`elevators`, splits 1/4). On the reported split it clips nothing—the largest scaled test value across all 14 datasets is 4.75. Some sweeps were run in parallel to a scratch directory and the result files copied in, so a few recorded commands name a temporary `--out` path; `reproduce.sh` writes to `results/` directly. Re-running the primary sweep from a clean clone reproduces every additive- and deep-model number to six significant figures except on California, the one uncapped dataset, where a later re-run under concurrent GPU load moved the deep KAN's nRMSE by 0.2%; the GROWKAN column, whose training includes the interaction screen and the purification penalty, moves by up to 0.7%, which on California crosses a third significant figure in two cells—the scripted check (`experiments/verify_reproduction.py`) reports exactly this and exits nonzero on it. GPU training is not bit-deterministic, and one workflow (`pykan`'s symbolic recipe) is not reproducible run to run even at fixed seeds, which §8 states. The B-spline basis is tested against `pykan` to $< 10^{-5}$, a pure-GAM EBM reproduces the California error of Agarwal et al. (2021) (0.55 against their 0.557), the three shape extractors are tested exact given a background sample—the SHAP permutation explainer is seeded independently of the model seed, so it contributes nothing to a comparison between retrainings—and the datasets (Table 8) are public.

AI USE STATEMENT

In this work we used a generative AI assistant (Claude), under author direction, for the following tasks in the categories ICLR requires be disclosed: *implementing methods* (the model, protocol and experiment code), *designing and giving feedback on experiments* (the controls of §5–§8 were proposed and critiqued in dialogue), *assisting in the writing of the proof* of Proposition 1 and *providing critical ingredients for proving it* (the strong-convexity step of part (i) and the exact-degeneracy construction of part (ii), both then checked by hand), *proposing and refining hypotheses* (objections to our own claims were raised and sharpened in dialogue—the matched-ϵ control of §4 exists be-

cause one such objection was pressed), and *interpreting results*. We did *not* use generative AI to generate synthetic datasets (the synthetic and Feynman data come from deterministic formulas in `src/kanlab/data/`), to formulate the paper's mathematical claims, to clean or reformat any dataset, or for translation. The remaining required-disclosure categories—developing a theoretical framework beyond the stated proposition, and qualitative or thematic data analysis—do not apply to this work.

We additionally used it in recommended-disclosure categories: drafting and editing the manuscript, creating figures, searching the literature, and identifying and formatting references.

We have reviewed all AI-assisted work, and the artifact is built so that the review is checkable rather than asserted: every number in the prose is tied to a committed result file by the test suite and must agree at every place it appears; every figure is regenerated by a `reproduce.sh` step; every result file the body cites carries a clean-tree provenance stamp; the proof was checked by hand; and every reference was resolved against arXiv, DOI or publisher metadata (`experiments/check_citations.py`). We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E. Hinton. Neural additive models: Interpretable machine learning with neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 4699–4711. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/251bd0442dfcc53b5a761e050f8022b8-Abstract.html.

Alejandro Almodóvar, Patricia A. Apellániz, Alba Garrido, Fernando Fernández-Salvador, Santiago Zazo, and Juan Parras. Interpretable clinical classification with kolmogorov-arnold networks, 2025. URL https://arxiv.org/abs/2509.16750v3.

David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018. URL https://arxiv.org/abs/1806.08049. arXiv:1806.08049.

Daniel W. Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020. doi: 10.1111/rssb.12377.

Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510, 1989. doi: 10.1214/aos/1176347115.

Mustafa Cavus, Jan N. van Rijn, and Przemysław Biecek. Beyond the single-best model: Rashomon partial dependence profile for trustworthy explanations in AutoML. In *Discovery Science (DS 2025)*, volume 16090 of *Lecture Notes in Computer Science*, pp. 445–459. Springer, 2025. doi: 10.1007/978-3-032-05461-6_29. URL https://doi.org/10.1007/978-3-032-05461-6_29.

Chun-Hao Chang, Sarah Tan, Ben Lengerich, Anna Goldenberg, and Rich Caruana. How interpretable and trustworthy are gams? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 95–105, 2021. doi: 10.1145/3447548.3467453. URL https://arxiv.org/abs/2006.06466.

Chun-Hao Chang, Rich Caruana, and Anna Goldenberg. NODE-GAM: Neural generalized additive model for interpretable deep learning. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.01613.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023. URL https://arxiv.org/abs/2305.01582v3.

Alexander D'Amour, Katherine Heller, Dan Moldovan, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022. URL http://jmlr.org/papers/v23/20-1335.html.

Salah A. Faroughi, Farinaz Mostajeran, Amirhossein Arzani, and Shirko Faroughi. Symbolic-KAN: Kolmogorov-arnold networks with discrete symbolic structure for interpretable learning, 2026. URL https://arxiv.org/abs/2603.23854v1.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL http://jmlr.org/papers/v20/18-760.html.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3681–3688, 2019. doi: 10.1609/aaai.v33i01.33013681.

Giles Hooker. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, 2007. doi: 10.1198/106186007X237892.

Bilal Khalid, Pedro Freire, Sergei K. Turitsyn, and Jaroslaw E. Prilepsky. Hardware-oriented inference complexity of kolmogorov-arnold networks, 2026. URL https://arxiv.org/abs/2604.03345v2.

Benjamin Lengerich, Sarah Tan, Chun-Hao Chang, Giles Hooker, and Rich Caruana. Purifying interaction effects with the functional ANOVA: An efficient algorithm for recovering identifiable additive models. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2402–2412, 2020. URL https://proceedings.mlr.press/v108/lengerich20a.html.

Ziyao Li. Kolmogorov-arnold networks are radial basis function networks, 2024. URL https://arxiv.org/abs/2405.06721.

Ziming Liu, Pingchuan Ma, Yixuan Wang, Wojciech Matusik, and Max Tegmark. Kan 2.0: Kolmogorov-arnold networks meet science, 2024. URL https://arxiv.org/abs/2408.10205.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2404.19756. arXiv:2404.19756.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pp. 623–631. ACM, 2013. doi: 10.1145/2487575.2487579. URL http://dx.doi.org/10.1145/2487575.2487579.

Charles T. Marx, Flavio du Pin Calmon, and Berk Ustun. Predictive multiplicity in classification. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6765–6774, 2020. URL https://proceedings.mlr.press/v119/marx20a.html.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability, 2019. URL https://arxiv.org/abs/1909.09223.

Sarah Polson and Vadim Sokolov. Kolmogorov GAM networks are all you need! *Entropy*, 27(6):593, 2025. doi: 10.3390/e27060593. URL https://doi.org/10.3390/e27060593.

Cynthia Rudin, Chudi Zhong, Lesia Semenova, Margo Seltzer, Ronald Parr, Jiachang Liu, Srikar Katta, Jon Donnelly, Harry Chen, and Zachery Boner. Position: Amazing things come from having many good models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42783–42795. PMLR, 2024. URL https://proceedings.mlr.press/v235/rudin24a.html.

Sidharth SS, Keerthana AR, Gokul R, and Anas KP. Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation, 2024. URL https://arxiv.org/abs/2405.07200v3.

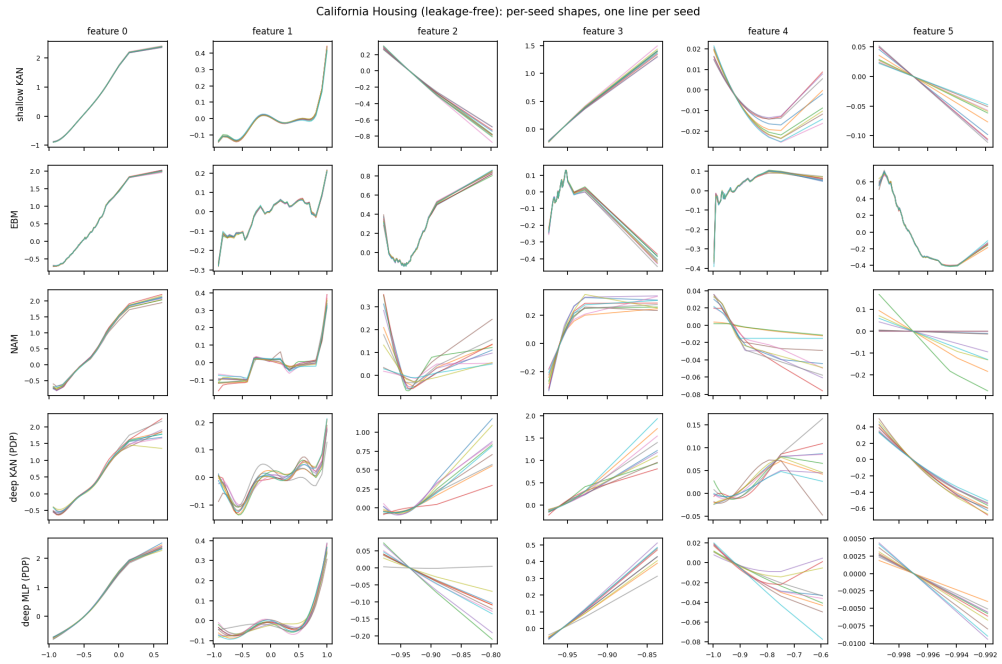

Figure 3: The core phenomenon, on California Housing under the leakage-free protocol (held-out split, train-only scaling, shapes on the train quantile grid; a separate ten-seed run by `experiments/regen_figures.py`): each panel overlays one model's learned shape for one feature across ten random seeds (centered). The spline GAM and EBM (top two rows) learn nearly identical shapes every time on this dataset; NAM (third row) is $6.5\times$ looser even here, the additivity-does-not-guarantee-it point of §4. Deep models read via partial dependence (bottom rows: deep KAN, deep MLP) spread widely across seeds: the "interpretation" changes with the random seed. Identifiability, not plottability, is what makes a shape trustworthy.

Charles J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13 (2):689–705, 1985. doi: 10.1214/aos/1176349548.

Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 4860–4871. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/hash/33a854e247155d590883b93bca53848a-Abstract.html`.

Runpeng Yu, Weihao Yu, and Xinchao Wang. Kan or mlp: A fairer comparison, 2024. URL `https://arxiv.org/abs/2407.16674v2`.

A Proposition 1 and Proof Sketch

Proposition 1 (Rashomon identifiability)**.** *Let L be population $\frac{1}{2}$MSE under a design μ with all $\mathbb{E}[x_j] = 0$, let $\mathcal{R}_\epsilon = \{f : L(f) - \inf L \le \epsilon\}$, and let S map an additive $f = \sum_j \phi_j$ to its vector of centered components in $\prod_j L^2(\mu_j)$. (i) For the additive class, if $L \circ S^{-1}$ is γ-strongly convex on the centered additive subspace in that norm (the* concurvity gap*; γ is the smallest eigenvalue of the covariance operator of the components, and is 0 under exact concurvity), then $D_j(\epsilon) \le \sqrt{8\epsilon/\gamma}$: shapes are pinned by the data. (ii) There exist designs with interaction and exact concurvity ($x_i = x_j$ a.s.), and two functions $f, f' \in \mathcal{R}_0$ of a class containing interactions, one of them additive, whose centered partial dependences of feature i are separated in $L^2(\mu_i)$ by a positive distance independent*

of ϵ. The same pair separates accumulated local effects, $\mathrm{ALE}_i(t) = t^2$ against $t^2/2$. (iii) Without degeneracy: let (x_1, x_2) be standard bivariate normal with correlation $0 \le \rho < 1$, target $f_0 = x_1 x_2$, and $f_\pm = f_0 \pm a(x_1 - x_2)^2$ with $a = \sqrt{\epsilon/6}/(1-\rho)$. Then $f_\pm \in \mathcal{R}_\epsilon$, their centered partial dependences of x_1 differ in $L^2(\mathcal{N}(0,1))$ by $2\sqrt{\epsilon/3}/(1-\rho)$, and their centered accumulated local effects by $2\sqrt{\epsilon/3}$: within one risk radius the PDP gap grows without bound as $\rho \to 1$, the ALE gap does not.

Two scope limits, stated because they bound what the proposition can be used for. It is about idealized classes, not trained networks; we treat it as the reason to look and the measurement below as the evidence. And (ii) is built on *exact* degeneracy ($x_i = x_j$ a.s.), so it is a limiting statement—what happens at finite concurvity is settled empirically by the ρ=0.8 sweep in §6, not by this argument. The ALE clause matters for a different reason: ALE is the standard remedy for partial dependence's extrapolation bias under correlated inputs, so a reader is entitled to ask whether our effect is merely that bias. It is not in measurement—the hierarchy survives under ALE on every dataset (§4). One dataset sits at the degenerate boundary of part (i): `elevators` has feature pairs at $|r| > 0.9999$, so $\gamma \approx 0$ and the bound is vacuous there—yet it is one of the confirmations; something else pins its additive shapes, or regularization does, and we flag it.

Proof sketch. (i) With $L = \frac{1}{2}$ MSE, γ-strong convexity gives $L(f) - L^\star \ge \frac{\gamma}{2}\|S(f) - S(f^\star)\|^2$ on the centered additive subspace, so $f \in \mathcal{R}_\epsilon$ implies $\|S(f) - S(f^\star)\| \le \sqrt{2\epsilon/\gamma}$; the triangle inequality between any two such f, f' gives $D_j \le 2\sqrt{2\epsilon/\gamma} = \sqrt{8\epsilon/\gamma}$ (so $C = \sqrt{8}$ in the body). (ii) Take $x_i = x_j$ almost surely (exact concurvity) and a target that equals x_i^2 on the support, so the Bayes-optimal predictor is attained by both the main-effect model $x_i \mapsto x_i^2$ and the interaction model $(x_i, x_j) \mapsto x_i x_j$ (they coincide on the support, hence both minimize the loss and lie in $\mathcal{R}_0$). Their *centered* feature-i partial dependences, $\mathrm{centered}(t^2)$ and $\mathrm{centered}(t\,\mathbb{E}[x_j])$ ($= 0$ when $\mathbb{E}[x_j]$=0), are separated in $L^2(\mu_i)$ by a fixed distance $c > 0$, so $D_i(0) \ge c$: partial dependence is not a function of the loss-equivalence class. Under this degenerate design the additive components x_i^2 and x_j^2 are likewise loss-equivalent, so (ii) shows non-uniqueness of a marginal summary at exact concurvity, not a property peculiar to depth; what separates the classes is (i), which needs $\gamma > 0$, and that is why the additive claim is conditioned on the design throughout. Note also that (i) uses the population risk and a norm on function components, while the experiments bound ϵ by a held-out RMSE or $1-$AUC range across seeds; for classification the strong-convexity argument does not transfer at all, since AUC is invariant to monotone rescaling of the score, and the classification results rest on the measurement alone. The same pair separates accumulated local effects, so the separation is not specific to partial dependence; on a degenerate support both depend on how f is extended off the diagonal, and the construction says nothing about non-degenerate designs, which the measurements address. With $\mathrm{ALE}_i(t) = \int_0^t \mathbb{E}[\partial_i f \mid x_i = z]\, dz$: for $f = x_i^2$, $\partial_i f = 2x_i$ gives $\mathrm{ALE}_i(t) = t^2$; for $f = x_i x_j$, $\partial_i f = x_j$ and $\mathbb{E}[x_j \mid x_i = z] = z$ under exact concurvity give $\mathrm{ALE}_i(t) = t^2/2$. The two differ by a factor of two, hence remain separated after centering. (iii) $x_1 - x_2 \sim \mathcal{N}(0, 2(1-\rho))$ has fourth moment $12(1-\rho)^2$, so $L(f_\pm) - L(f_0) = \frac{1}{2}a^2\,\mathbb{E}[(x_1 - x_2)^4] = 6a^2(1-\rho)^2 = \epsilon$. With x_2 drawn from its marginal, $\mathrm{PD}_1(f_\pm)(t) = \pm a(t^2 + 1)$, centered $\pm a(t^2 - 1)$, whose $L^2(\mathcal{N}(0,1))$ norm is $a\sqrt{2}$; the gap is $2\sqrt{2}\,a$. For ALE, $\partial_1 f_\pm = x_2 \pm 2a(x_1 - x_2)$ and $\mathbb{E}[x_2 \mid x_1 = z] = \rho z$ give $\mathrm{ALE}_1(t) = (\rho/2 \pm a(1-\rho))t^2$, a gap of $2\sqrt{2}\,a(1-\rho)$ after centering. Substituting a gives both expressions; `tests/test_proposition.py` checks them by Monte Carlo. The construction is one example: it shows PDP and ALE separate at finite concurvity, not that ALE is stable for trained networks, which the measurements address. □

An exploratory relation, not a rate test. If the class's ϵ scaled as $1/n$, the bound $D_j \le C\sqrt{\epsilon/\gamma}$ would suggest an $n^{-1/2}$ stabilization rate for an additive model's shapes. We cannot measure the class's ϵ; what we can measure is the learner's spread of held-out loss across retrainings at each n, and we report how it moves alongside shape disagreement as an exploratory relation only.

On `fried` the additive model's loss spread falls as $n^{-1.24}$ and its shape disagreement as $n^{-0.53}$, roughly the $n^{-1/2}$ the bound would suggest, which is consistent with the argument but does not test it.

Two qualifications. First, what the data separates is ϵ, not the rate law. (Throughout, "Rashomon set" here means the observed seed spread of held-out loss, a property of the learner rather than of the class.) On every dataset whose spread shrinks at all the additive model's collapses far faster than the deep model's—$n^{-1.61}$ against $n^{-0.17}$ on `house16`, $n^{-1.24}$ against $n^{-0.61}$ on `fried`, $n^{-2.24}$ against $n^{-0.53}$ on `elevators`, $n^{-2.01}$ against $n^{-0.75}$ on `magic`—but shape decay does *not* track $\sqrt{\epsilon}$ quantitatively: the deviation reaches 0.80 in the exponent on `elevators`. Part (i) is an upper bound, not a rate law, and with three or four sizes per dataset we cannot test it as one. We report the asymmetry, which is large and consistent, and not a fitted exponent. It is also not an artifact of how ϵ is summarized: replacing the seed standard deviation by the range or the interquartile range of held-out loss leaves the additive model's Rashomon set collapsing faster than the deep model's on every dataset where either collapses at all.

Second, `diamonds` is not the counterexample it appears. Its ϵ *grows* with n ($n^{+0.16}$ additive, $n^{+0.49}$ deep) and its shapes correspondingly fail to stabilize, so the two move together there as well; nothing decays because the Rashomon set is not shrinking.

Third, and most important for reading §8: the symbolic-workflow result lies *outside* the proposition's scope. Part (ii) bounds nothing about D_j for a deep model, but D_j is a distance between *functions*; what fails to reproduce across seeds on Feynman is the *expression*, while the functions agree to correlation 0.95–1.00 at ϵ small enough that the RMSE is 0.009. That is non-identifiability of the representation at $\epsilon \to 0$, a stronger and different statement than the proposition makes, and we claim no theory for it. One structural reason it should be expected can be stated, though. The map from expressions to functions is many-to-one even at $\epsilon = 0$, and a two-layer KAN, whose edge primitives are univariate, has more than one expression tree for a product: $x_1 x_2 = \exp(\log x_1 + \log x_2)$ on one hidden node, or $x_1 x_2 = \frac{1}{4}\left[(x_1+x_2)^2 - (x_1-x_2)^2\right]$ on two, the latter valid on negative inputs where the former is not. The primitive assignment `auto_symbolic` returns is therefore a choice among trees computing the same function, made edge by edge from a fit-plus-complexity score. Convergence of the *function* across seeds therefore does not imply convergence of the *tree*, which is what Table 5 shows; identifiability of an expression is well-posed only relative to a canonical form, which neither `pykan` nor the paper's skeleton metric imposes.

B Edge Functions vs Partial Dependence

A deep KAN's layer-0 edge functions are reconstructed from the parameters, $\phi_{i\to j}(t) = \mathrm{base}_{j,i}\,\mathrm{SiLU}(t) + \sum_m c_{j,i,m} B_m(t)$, on each feature's own quantile grid, and compared across seeds by centered correlation. The forward pass cannot isolate a single edge, because a node sums over all of its incoming edges, so the reconstruction is done from the weights directly.

Three readings are reported per dataset (`experiments/edge_vs_pdp.py`). *As indexed* compares $\phi_{i\to j}$ to the same (i, j) in another seed; this is what a reader comparing two published edge plots would be doing, and it is close to zero because hidden-unit permutation makes the index meaningless. *Matched* first solves a Hungarian assignment per input feature, maximising total correlation between the two seeds' hidden units — an oracle that quotients out the permutation symmetry, though not the rescaling one. *PDP* is the paper's headline read on the same trained models.

The ordering as-indexed $\ll$ matched $<$ PDP is what justifies reporting partial dependence: it is the most favourable well-posed reading of a deep KAN available, so the paper's negative result is not obtained by choosing a fragile estimator.

C GrowKAN: Construction and Frontier

To check that the standard is meetable, we build for it: an additive-first model that screens interactions on the train residual, adds only the top-k pairs as separate blocks, and penalizes them for absorbing main-effect structure—the functional-ANOVA purity constraint of Lengerich et al. (2020), imposed during training rather than recovered afterwards. The accuracy–identifiability curve it traces is shown on `airfoil` (Fig. 6) and California (Fig. 7), and the penalty's weight is swept in Fig. 5. The frontier verdict (3/14) and the reason for it are in §9; an earlier draft reported $8/14$ over a candidate set omitting the interaction-GAMs, a number that was not falsifiable and is withdrawn.

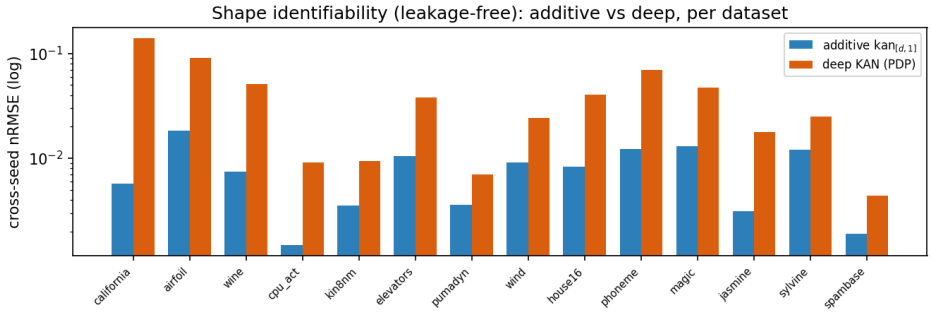

Figure 4: *Visual form of Tables 1 and 2.* Cross-seed shape disagreement (nRMSE, lower is more identifiable) under the leakage-free protocol, on all 14 corrected datasets. The additive kan$_{[d,1]}$ (blue) has lower disagreement than the deep KAN PDP (orange) on *every* dataset—the additive<deep hierarchy of Tables 1, 2, visualized. (Deep MLP, omitted here for clarity, is not reliably more or less stable than deep KAN across datasets—§5.)

The identifiability axis also separates the two interaction-GAMs in a way no accuracy table can. NodeGAM has no stored main-effect terms, so both models are read by partial dependence of the full predictor, which folds each model's interaction terms' marginal means in, and both are normalized by their own output scale. On that like-for-like reading NodeGAM's main effects are less identifiable than EBM-GA2M's on **14/14** datasets: 1.0–12.4× on regression (median 4.1×) and 2.4–5.4× on classification (median 4.4×). These are point estimates without a seed interval; the closest, `wind` at 1.01×, is not resolved, so "14/14" is a count of point orderings. An earlier version reported 17.9–56.7× on classification from a comparison that read EBM's native terms against NodeGAM's PDP and left NodeGAM's logits unnormalized; both defects inflated the gap. EBM's own PDP is 1.0–3.4× looser than its native terms, which is the cost of folding interactions in. GrowKAN is scored on its native main effects, the same object as EBM's native terms, so the frontier comparison uses those; its verdict (3/14) is unchanged. NodeGAM dominates nothing, for a measured reason.

D Symbolic Workflow: Settings and Their Justification

The prune-and-symbolify experiment uses `pykan` 0.2.8 (pinned in `requirements.txt`), width $[d, 5, 1]$, `grid`$= 5$, k=3, 50 LBFGS steps per stage with λ=0.01 sparsification, and the five-stage sequence the `pykan` tutorials prescribe: sparsified training, `prune`, *retrain*, `auto_symbolic`, *refit* with the grid frozen. Both retraining stages are load-bearing—pruning alone takes a toy fit from RMSE 0.03 to 0.31, and the post-symbolic refit must not touch the grid because no spline activations remain. The sequence recovers `pykan`'s canonical $\exp(\sin \pi x_1 + x_2^2)$ example to the digit, so the environment is not the cause of what follows.

One setting is non-default and must be disclosed. `auto_symbolic` selects each edge's primitive by a weighted sum of fit and complexity, `weight_simple`, whose default of 0.8 collapses every real-data model we tried to a constant: on California an edge that sin fits at r^2=0.98 is still assigned the zero primitive, because the complexity term dominates, and the resulting formula is the target mean on every seed. At 0.0 the fit overflows (tan, arctanh). 0.3 is the largest value at which the workflow produces a non-constant formula, so it is run there—at its best, not its default. The default's collapse is itself worth knowing for anyone applying the workflow to real data.

Budget matters for recovery and not for reproducibility. Re-running the four Feynman equations with 200 LBFGS steps per stage instead of 50 (Table 5, second block) leaves the picture unchanged on three of them—no two seeds share a structure on $x_1(x_2-x_3)x_4$, $\arcsin(x_1 \sin x_2)$ or the Gaussian, and the majority form on $x_1/\sqrt{1-x_2^2}$ persists—but one seed of five now lands within 10^{-3} of $x_1(x_2-x_3)x_4$, as a product of affine factors, at RMSE 0.0002. The 50-step claim that the workflow never recovers the equation was therefore a statement about the budget; what survives at both budgets is that the seeds do not agree with each other.

Generated inline numbers. Every inline number that comes from a result file rather than a table is typeset from `numbers.tex`, which `gen_tables.py` writes: the edge agreements of §3 (as-indexed +0.01, Hungarian-matched +0.75, partial dependence +0.94, over 9 regression datasets), the MLP/KAN median ratio and the leave-one-out correlation range of §4–5, the capacity-per-basis minimum, the SHAP count, the run-to-run spread and finite-formula range, the distances to the generating equation and the 200-step recovery count, and the knot-refinement counts of §8. They are generated for the same reason the tables are—a number typed into a sentence has no protection against drifting from its log—and a test asserts each matches its file.

Superseded results. `results/discovery` holds the earlier two-screen validation (planted pairs on synthetic data; our screen against EBM's on six real datasets) behind Table 26. It is superseded by the five-screen, fourteen-dataset experiment of §7 and is retained only because the synthetic planted-pair recall it reports is not repeated there; nothing in the body rests on it.

E Validating the Operation Counter

The cost claim rests on counting work that standard profilers skip, so the counter needs validating. Comparing it against another profiler would not do that: every standard tool counts the matmul/conv family by design, which is precisely the omission at issue. `torch.utils.flop_counter` on the same deep KAN returns exactly our matmul-only column (up to its 2-FLOPs-per-MAC convention), confirming the blind spot is shared rather than peculiar to our implementation.

We validate instead against a count derived by hand. For `efficient-kan`'s `b_splines` with grid size G and spline order K, the knot vector holds $G + 2K + 1$ entries, the indicator step emits width $G + 2K$, and each of the K Cox–de Boor iterations narrows the width by one while applying seven batched elementwise operations (two subtractions, two divisions, two multiplications, one addition). Per input feature per sample that is $3(G{+}2K) + 7\sum_{k=1}^{K}(G{+}2K{-}k)$ operations, the indicator step contributing three, not four, because its trailing dtype cast is a conversion and not arithmetic. For $G{=}5$, $K{=}3$ this predicts 222; the counter measures 222.2, a 0.1% agreement (`experiments/flop_validation.py`).

Two consequences are worth separating. That most of a deep KAN's operations are invisible to matmul-only counting is robust—81% on `california` and 69% on `pumadyn` under a MAC convention, 67% and 53% under a strict FLOP one, since counting a multiply–accumulate twice doubles the matmul half and leaves the spline recursion alone—and it is the finding we rest on. *How much* more expensive the remainder makes a KAN is not robust at all: under a MAC convention the ratio is 1.4–$2.4\times$, under a strict FLOP convention 0.84–$1.61\times$. Neither convention is wrong, and each favours the architecture whose work it under-weights, so the paper reports both. A third caveat is the implementation. Khalid et al. (2026) count the same de Boor recursion per edge for hardware inference and reach the opposite emphasis: a sparsity-aware implementation evaluates only the $k{+}1$ active basis functions, so a count that scales with the grid size—ours, and every GPU FLOP count they survey— *overestimates* a KAN's cost on such hardware. Our numbers are for the dense GPU implementation we trained and timed (`efficient-kan`); the blind spot is a property of the profiler, the ratio a property of the implementation.

F Full Results

All tables are generated directly from the committed result logs by a script, so every number traces to data. Those the body argues from are listed first.

Cited in the body. Tables 1 and 2 are the primary leakage-free results; Table 21 the multi-split robustness and sign tests, Table 22 held-out accuracy with 95% CIs across the five splits, Table 23 the PDP-vs-ALE robustness, Table 24 the equal-accuracy sensitivity, Table 7 the same measurement at ten seeds, Table 13 the hierarchy at a common Rashomon radius, Table 10 the hierarchy under SHAP dependence curves, Table 6 the dataset-size sweep, Table 18 the knot-placement control, Table 12 the knot-refinement control, Table 16 the cross-seed vs cross-method comparison for discovered interactions, Table 19 the sparsification workflow, Table 5 the prune-and-symbolify workflow with its PySR control, Table 9 the run-to-run repeats, Table 3 the capacity sweep behind the definition of

Table 3: **No non-additive capacity closes the gap.** Cross-seed PDP nRMSE of six non-additive configurations as a multiple of the additive model's, on every dataset at ten seeds. The most stable non-additive configuration is still $1.7\times$ the additive model at minimum and $2.6\times$ at the median, and the ordering holds on 14/14. Capacity acts differently on the two architectures: from smallest to largest, the deep MLP's disagreement changes by a median $1.30\times$ (worse on 11/14) and the deep KAN's by $0.97\times$ (worse on 6/14). Generated from `results/capacity`.

dataset	additive nRMSE	deep KAN, hidden [16]	deep KAN, hidden [32]	deep KAN, hidden [64, 32]	deep MLP, hidden [64, 64]	deep MLP, hidden [128, 128]	deep MLP, hidden [256, 128, 64]
california	0.0070	20.4	18.2	14.9	16.6	20.5	33.8
airfoil	0.0218	3.5	2.7	1.9	2.9	3.9	5.0
wine	0.0064	7.1	7.5	5.0	3.5	5.1	6.5
cpu_act	0.0013	6.2	6.5	7.0	4.2	5.6	6.5
kin8nm	0.0032	3.0	2.6	3.2	3.3	4.1	3.8
elevators	0.0105	3.3	2.9	3.2	3.9	4.0	4.9
pumadyn	0.0037	1.7	1.8	2.7	3.2	3.0	2.1
wind	0.0079	2.8	2.8	3.1	2.6	3.2	3.1
house16	0.0099	3.5	3.3	3.1	3.5	3.6	3.5
phoneme	0.0273	2.4	2.7	2.6	2.4	2.7	2.7
magic	0.0186	2.5	2.7	2.4	2.7	3.0	3.7
jasmine	0.0034	4.6	4.4	3.0	7.1	8.3	11.4
sylvine	0.0113	2.6	2.6	3.2	2.7	3.0	2.2
spambase	0.0018	2.6	2.8	2.3	3.2	5.4	6.6

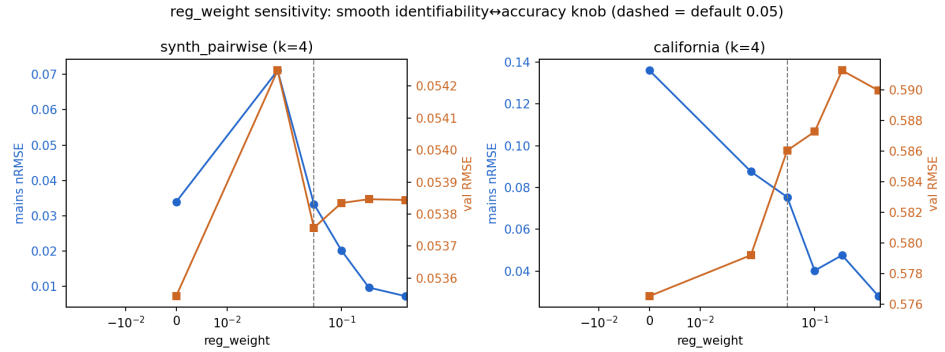

Figure 5: Regularization-weight sensitivity at $k{=}4$ (within-sample). The penalty trades identifiability against accuracy: on synthetic data (left) accuracy is flat while nRMSE falls; on California (right) identifiability improves $5\times$ at a 2.3% RMSE cost. The trend is directional but not strictly monotone—each panel has one bump (synthetic at weight 0.02, California at 0.2), visible above. The default (0.05, dashed) is conservative.

"deep", Table 4 the validation-selected configurations, Table 17 the cost triple, Table 27 cross-basis generality and Table 11 capacity per basis, Table 25 the Feynman evaluation, Tables 20 and 35 the interaction-order diagnosis and its within-sample cross-check, Table 34 the head-to-head with the interaction-GAM competitors, Table 14 the binary datasets our primary suite does not use, Table 15 every class read by one explanation object, and Table 8 the datasets as used.

Reference material, not load-bearing. The remainder record analyses the body does not rely on, retained so a reader can check them: the within-sample identifiability suite that preceded the leakage-free protocol (Tables 30, 31, 32, 33), the interaction-discovery validation on planted pairs (Table 26), and two sensitivity sweeps—screened-versus-random pairs (Table 28) and grid density (Table 29). No claim in the paper rests on any of them.

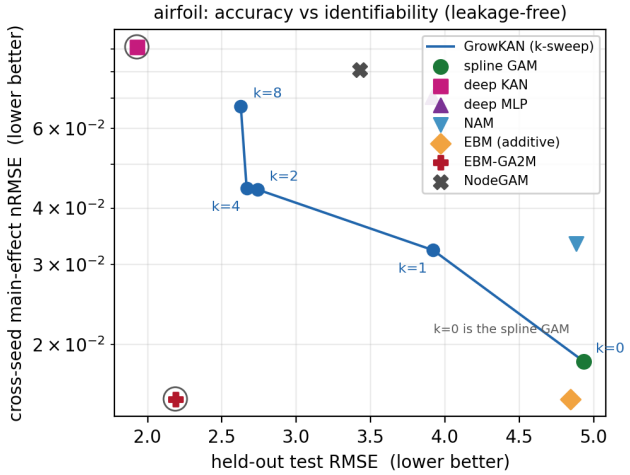

Figure 6: **Accuracy–identifiability plane on `airfoil`** (leakage-free: held-out test error, pairs screened on the train residual only, 5 seeds). Circled points are Pareto-optimal. Growing interactions traces a curve from the additive model (k=0, which *is* the spline GAM, hence the coincident markers) toward deep-KAN accuracy, and against the additive and deep baselines alone the whole curve is on the frontier. It is not once EBM-GA2M is included: that model sits below and to the left of every GrowKAN point, and the frontier is {deep KAN, EBM-GA2M}. This is the picture behind the $3/14$ of §9, and `airfoil` is one of the 11 where GrowKAN is dominated—we show it instead of a dataset where we win, because `airfoil` is also where the additive control attributes 97% of the gap to genuine interactions (versus 1% on California, Fig. 7), i.e. the case most favourable to growing them.

Table 4: **Tuning does not change the ordering, and widens the KAN–MLP gap.** For each model a learning-rate $\times$ width grid (3×3; 3 for the additive model, which has no width) is run and the configuration with the best *validation* metric is selected – never by test, never by nRMSE. The selected configuration's held-out error and cross-seed nRMSE are shown. The additive model stays the most identifiable on 14/14 as a sign count, though
textttkin8nm and
textttwind sit inside 1 SE; the tuned KAN is more stable than the tuned MLP on 12/14 (median ratio 0.78), because selection favours large architectures and the MLP's partial dependence degrades with size while the KAN's does not. Generated from `results/tune`.

dataset	model	selected	test	nRMSE
california	kan_shallow	lr=0.01	0.6537	0.00855±0.00217
	kan	lr=0.003, [64, 32]	0.4913	0.104±0.008
	mlp	lr=0.003, [256, 128, 64]	0.5098	0.199±0.064
airfoil	kan_shallow	lr=0.01	4.927	0.0283±0.0055
	kan	lr=0.003, [64, 32]	1.482	0.0389±0.0048
	mlp	lr=0.01, [256, 128, 64]	1.825	0.0872±0.0282
wine	kan_shallow	lr=0.01	0.7406	0.00949±0.00148
	kan	lr=0.003, [64, 32]	0.7147	0.0303±0.0036
	mlp	lr=0.003, [256, 128, 64]	0.6988	0.0379±0.0045
cpu_act	kan_shallow	lr=0.01	2.879	0.00175±0.00019
	kan	lr=0.003, [64, 32]	2.429	0.00854±0.00111
	mlp	lr=0.01, [256, 128, 64]	2.413	0.00873±0.00102
kin8nm	kan_shallow	lr=0.01	0.1944	0.00655±0.00052
	kan	lr=0.001, [64, 32]	0.06563	0.00684±0.00054
	mlp	lr=0.001, [256, 128, 64]	0.0637	0.00921±0.00163
elevators	kan_shallow	lr=0.01	0.002383	0.0133±0.0014
	kan	lr=0.003, [64, 32]	0.001859	0.0369±0.0062
	mlp	lr=0.003, [256, 128, 64]	0.001906	0.0504±0.0107
pumadyn	kan_shallow	lr=0.01	0.034	0.00644±0.00043
	kan	lr=0.01, [64, 32]	0.02195	0.00963±0.00050
	mlp	lr=0.003, [256, 128, 64]	0.02223	0.00749±0.00043
wind	kan_shallow	lr=0.01	3.02	0.0161±0.0058
	kan	lr=0.003, [16]	2.981	0.0242±0.0041
	mlp	lr=0.003, [256, 128, 64]	2.928	0.0272±0.0008
house16	kan_shallow	lr=0.01	4.049e+04	0.0135±0.0022
	kan	lr=0.01, [64, 32]	3.676e+04	0.0289±0.0019
	mlp	lr=0.003, [128, 128]	3.97e+04	0.0353±0.0063
phoneme	kan_shallow	lr=0.01	0.1277	0.0163±0.0025
	kan	lr=0.01, [64, 32]	0.0716	0.0830±0.0148
	mlp	lr=0.01, [256, 128, 64]	0.08044	0.210±0.125
magic	kan_shallow	lr=0.01	0.1094	0.0143±0.0020
	kan	lr=0.01, [32]	0.07619	0.0549±0.0049
	mlp	lr=0.01, [256, 128, 64]	0.07717	0.0720±0.0172
jasmine	kan_shallow	lr=0.003	0.1948	0.00313±0.00041
	kan	lr=0.003, [64, 32]	0.2009	0.00963±0.00140
	mlp	lr=0.01, [256, 128, 64]	0.1847	0.0956±0.0229
sylvine	kan_shallow	lr=0.01	0.01922	0.0164±0.0006
	kan	lr=0.01, [16]	0.01893	0.0299±0.0045
	mlp	lr=0.003, [256, 128, 64]	0.02122	0.0188±0.0016
spambase	kan_shallow	lr=0.003	0.02433	0.00191±0.00028
	kan	lr=0.01, [16]	0.02319	0.00910±0.00152
	mlp	lr=0.003, [256, 128, 64]	0.02947	0.00934±0.00170

Table 5: **KAN's own symbolic workflow is not reproducible either.** `pykan`'s recipe – sparsified training, prune, retrain, symbolic regression on every surviving edge, refit – run once per seed on the leakage-free split, at the complexity weight that yields non-constant formulas (the default collapses every real-data model to a constant). *ok*: seeds on which the workflow returned a finite formula at all. Agreement is across those seeds: pruned architecture, per-edge primitive, exact formula string, *skeleton* (the expression with every constant replaced, i.e. its structure), and correlation of the formulas' held-out predictions. The last is high while the formulas share almost nothing – the Rashomon set, written out as equations. The Feynman rows are the setting KAN advertises for symbolic recovery. *vs truth*: range over seeds of $\max|\hat{f} - f^\star|$ on the input domain (standardized target), with $f^\star$ the generating equation mapped into the model's coordinates; 0 to 10^{-3} would mean the equation was recovered. The last block is PySR (Cranmer, 2023) on the same rows, split and coordinates, scored the same way, single-threaded so a seed fixes its result. Agreement columns are over the *ok* seeds only. Generated from `results/symbolic_workflow` and `results/pysr_control`.

dataset	ok	arch	primitive	formula	skeleton	pred. corr	sym./pruned RMSE	vs truth
california	2/5	0.00	0.12	0.00	0.00	0.91	1.16×	–
airfoil	3/5	1.00	0.09	0.00	0.00	0.91	1.62×	–
kin8nm	3/5	0.33	0.17	0.00	0.00	0.97	0.99×	–
feynman-I.6.2	5/5	0.40	0.18	0.00	0.00	0.99	4.93×	0.42–0.83
feynman-I.26.2	3/5	0.33	0.22	0.00	0.00	1.00	9.88×	0.07–0.10
feynman-I.15.3t	5/5	1.00	0.87	0.00	0.60	1.00	4.58×	0.04–0.04
feynman-II.2.42	4/5	0.50	0.37	0.00	0.00	1.00	1.11×	0.01–0.59
KAN workflow at 200 LBFGS steps per stage (four times the budget above)								
I.6.2 (200)	4/5	1.00	0.17	0.00	0.00	0.98	8.24×	0.64–1.15
I.26.2 (200)	5/5	1.00	0.22	0.00	0.00	1.00	17.97×	0.06–0.16
I.15.3t (200)	5/5	1.00	0.87	0.00	0.60	1.00	4.58×	0.04–0.04
II.2.42 (200)	5/5	0.60	0.38	0.00	0.00	1.00	1.09×	0.00–0.68
PySR on the same data (control); RMSE column is held-out test RMSE								
PySR I.6.2	5/5	–	–	0.00	0.00	1.00	0.086	0.34–0.78
PySR I.26.2	5/5	–	–	0.00	0.00	1.00	0.007	0.03–0.07
PySR I.15.3t	5/5	–	–	0.00	0.00	1.00	0.006	0.01–0.08
PySR II.2.42	5/5	–	–	0.00	0.00	1.00	0.021	0.00–0.35

Table 6: **Is the hierarchy a small-data artifact?** The identical leakage-free protocol at several dataset sizes on `diamonds`, `fried`, `house16`, `elevators`, `magic` (ten seeds; `magic` is classification, metric $1-$AUC). Rows failing the 2% meaningful-gap criterion used throughout are marked †, since there the two models are doing the same thing and the ratio compares nothing. The ordering holds on 17/17 cells.

dataset	n	GAM	deep	additive nRMSE	deep nRMSE	ratio
diamonds†	6,000	1.48e+03	1.47e+03	0.0198±0.0027	0.0235±0.0021	1.2
diamonds	12,000	1.35e+03	1.31e+03	0.0202±0.0030	0.0517±0.0062	2.6
diamonds	25,000	1.43e+03	1.39e+03	0.00819±0.00122	0.0564±0.0218	6.9
diamonds†	50,000	1.4e+03	1.38e+03	0.0342±0.0102	0.0500±0.0181	1.5
fried	6,000	1.6	1.01	0.00752±0.00029	0.00855±0.00019	1.1
fried	12,000	1.67	1.01	0.00504±0.00046	0.00536±0.00022	1.1
fried	25,000	1.69	1.01	0.00253±0.00017	0.00497±0.00030	2.0
fried	50,000	1.68	1.01	0.00270±0.00012	0.00529±0.00028	2.0
house16	6,000	4.39e+04	4.13e+04	0.0150±0.0018	0.0248±0.0026	1.7
house16	11,000	3.79e+04	3.63e+04	0.00328±0.00036	0.0363±0.0018	11.0
house16	22,000	3.96e+04	3.41e+04	0.00255±0.00022	0.0359±0.0019	14.1
elevators	6,000	0.0024	0.00196	0.0125±0.0014	0.0334±0.0031	2.7
elevators	11,000	0.00244	0.00198	0.00944±0.00184	0.0253±0.0025	2.7
elevators	16,599	0.00237	0.0019	0.00914±0.00099	0.0275±0.0028	3.0
magic	6,000	0.118	0.0959	0.0158±0.0028	0.0511±0.0095	3.2
magic	12,000	0.11	0.0777	0.0186±0.0018	0.0472±0.0030	2.5
magic	19,020	0.0962	0.0691	0.00422±0.00063	0.0673±0.0049	16.0

Table 7: **The primary measurement at ten seeds**, against the five-seed version the paper reports. Cross-seed nRMSE with leave-one-seed-out jackknife SEs. The additive<deep ordering is unchanged on every dataset; the ratio is not, which is why the paper claims the former and not the latter. Generated from `results/corrected_seeds10`.

	additive nRMSE		deep nRMSE		ratio	
dataset	5 seeds	10 seeds	5 seeds	10 seeds	5 seeds	10 seeds
california	0.00574±0.00115	0.00697±0.00073	0.141±0.016	0.142±0.013	24.5	20.4
airfoil	0.0183±0.0054	0.0218±0.0038	0.0908±0.0133	0.0761±0.0061	5.0	3.5
wine	0.00745±0.00190	0.00641±0.00119	0.0509±0.0173	0.0456±0.0058	6.8	7.1
cpu_act	0.00147±0.00020	0.00129±0.00014	0.00915±0.00260	0.00796±0.00123	6.2	6.2
kin8nm	0.00356±0.00035	0.00316±0.00020	0.00933±0.00078	0.00952±0.00042	2.6	3.0
elevators	0.0105±0.0012	0.0105±0.0009	0.0380±0.0093	0.0343±0.0048	3.6	3.3
pumadyn	0.00362±0.00038	0.00368±0.00026	0.00697±0.00036	0.00642±0.00025	1.9	1.7
wind	0.00904±0.00078	0.00788±0.00086	0.0242±0.0041	0.0219±0.0015	2.7	2.8
house16	0.00835±0.00087	0.00988±0.00183	0.0407±0.0074	0.0344±0.0034	4.9	3.5
phoneme	0.0122±0.0061	0.0273±0.0086	0.0698±0.0113	0.0668±0.0058	5.7	2.4
magic	0.0131±0.0025	0.0186±0.0018	0.0473±0.0067	0.0472±0.0030	3.6	2.5
jasmine	0.00313±0.00041	0.00339±0.00018	0.0176±0.0024	0.0156±0.0016	5.6	4.6
sylvine	0.0121±0.0018	0.0113±0.0006	0.0248±0.0047	0.0293±0.0026	2.1	2.6
spambase	0.00191±0.00028	0.00185±0.00018	0.00442±0.00040	0.00485±0.00032	2.3	2.6

Table 8: **Datasets as used.** n is the row count after preprocessing: OpenML sources with more than 12,000 rows are subsampled uniformly to 12,000 (seed 0; the scale sweep of Table 6 raises this cap), California uses all 20,640 rows. d counts numeric features: non-numeric columns are dropped (`diamonds` loses cut, color and clarity), rows with a missing value are dropped, and the classification target is the lexically last class label. Splits are 60/20/20 train/val/test (stratified for classification), features min-max scaled to $[-1, 1]$ on the train split. Generated from the result files (`n_train+n_val+n_test`, `d`).

dataset	task	n	d	source
california	regression	20,640	8	sklearn `fetch_california_housing`
airfoil	regression	1,503	5	OpenML `airfoil_self_noise` v1
wine	regression	6,497	11	OpenML `wine_quality` v1
cpu_act	regression	8,192	21	OpenML `cpu_act` v1
kin8nm	regression	8,192	8	OpenML `kin8nm` v1
elevators	regression	12,000	18	OpenML id 216
pumadyn	regression	8,192	32	OpenML `pumadyn32nh` v1
wind	regression	6,574	14	OpenML `wind` v1
house16	regression	12,000	16	OpenML `house_16H` v1
phoneme	classification	5,404	5	OpenML `phoneme` v1
magic	classification	12,000	10	OpenML `MagicTelescope` v1
jasmine	classification	2,984	8	OpenML `jasmine` v1
sylvine	classification	5,124	20	OpenML `sylvine` v1
spambase	classification	4,601	57	OpenML `spambase` v1
diamonds	regression	50,000	6	OpenML `diamonds` v1 (scale sweep, largest size)
fried	regression	40,768	10	OpenML `fried` v1 (scale sweep, largest size)

Table 9: **Run-to-run variation the seed intervals do not cover.** The symbolic workflow and the sparsification sweep re-run with identical seeds and code from a clean commit; GPU training is not bit-deterministic, and a five-seed dispersion statistic inherits that. Symbolic: seeds returning a formula, skeleton agreement and prediction correlation per run. Sparsification: the deep KAN's PDP disagreement unregularized and at its best weight, per run, with the spread $(\max - \min)/\mathrm{mean}$. Conclusions in the body are stated as ranges over these runs where they move. Generated from `results/*/repeats`.

symbolic	ok r1	ok r2	ok r3	skel. r1	skel. r2	skel. r3	corr r1	corr r2	corr r3
california	2/5	4/5	2/5	0.00	0.00	0.00	0.91	0.94	0.96
airfoil	3/5	3/5	1/5	0.00	0.00	–	0.91	0.96	–
kin8nm	3/5	5/5	5/5	0.00	0.00	0.00	0.97	0.95	0.95
feynman-I.6.2	5/5	3/5	5/5	0.00	0.00	0.00	0.99	0.99	0.98
feynman-I.26.2	3/5	4/5	3/5	0.00	0.00	0.00	1.00	1.00	1.00
feynman-I.15.3t	5/5	4/5	4/5	0.60	0.50	0.50	1.00	1.00	1.00
feynman-II.2.42	4/5	5/5	5/5	0.00	0.00	0.00	1.00	1.00	1.00
sparsification	w=0 r1	w=0 r2	w=0 r3	best r1	best r2	best r3	spread (w=0 / best)		
california	0.141	0.141	0.141	0.050	0.059	0.052	0% / 17%		
airfoil	0.091	0.091	0.091	0.062	0.063	0.063	0% / 3%		
kin8nm	0.009	0.009	0.009	0.013	0.013	0.014	0% / 5%		

Table 10: **The hierarchy under SHAP dependence curves.** Interventional SHAP values (permutation explainer, independent background of 100 train rows) on 512 train rows, binned by feature value onto the shared quantile grid – the curve a SHAP dependence plot draws – then scored exactly as PDP: cross-seed nRMSE over seeds, same split and models as Tables 1 and 2. For the additive model the SHAP curve is its native shape up to binning (compare the first two columns), which validates the extractor. The additive<deep ordering holds on 14/14. Generated from `results/shap_extractor`.

	additive		deep KAN		SHAP	
dataset	native	SHAP	PDP	SHAP	deep/additive	holds
california	0.0057	0.0063	0.1443	0.1248	19.9×	yes
airfoil	0.0183	0.0177	0.0906	0.0803	4.5×	yes
wine	0.0075	0.0075	0.0487	0.0476	6.3×	yes
cpu_act	0.0015	0.0014	0.0093	0.0065	4.8×	yes
kin8nm	0.0036	0.0036	0.0112	0.0184	5.2×	yes
elevators	0.0105	0.0110	0.0378	0.0366	3.3×	yes
pumadyn	0.0036	0.0036	0.0067	0.0087	2.4×	yes
wind	0.0090	0.0085	0.0231	0.0221	2.6×	yes
house16	0.0084	0.0078	0.0463	0.0349	4.5×	yes
phoneme	0.0122	0.0119	0.0772	0.0761	6.4×	yes
magic	0.0131	0.0131	0.0481	0.0462	3.5×	yes
jasmine	0.0031	0.0024	0.0174	0.0149	6.1×	yes
sylvine	0.0121	0.0118	0.0254	0.0254	2.2×	yes
spambase	0.0019	0.0015	0.0046	0.0042	2.7×	yes

Table 11: **Capacity trend per basis.** The deep model of each basis (spline KAN $[d, h, 1]$; RBF and Chebyshev: basis-embedded MLP of hidden width h) at three widths, 8 seeds, same protocol as Table 27. *add.* is the additive model's cross-seed nRMSE for that basis; the h columns are the deep PDP's nRMSE divided by it, so 1 means capacity closed the gap *within that basis*. Global polynomial (Chebyshev) shapes are themselves several times less reproducible than spline shapes, which is where the sub-1 entries come from; against the spline additive model, the paper's reference, no deep model of any basis or width comes within 1.7×. Generated from `results/crossbasis_capacity`.

	spline				rbf				cheb			
dataset	add.	h=16	h=64	h=256	add.	h=16	h=64	h=256	add.	h=16	h=64	h=256
california	0.0069	22.0	22.2	25.7	0.0039	15.4	11.4	9.4	0.0084	8.8	6.8	4.7
cpu_act	0.0013	6.2	6.6	6.3	0.0039	1.4	1.5	1.4	0.0100	1.5	1.2	0.8
kin8nm	0.0032	2.9	2.5	2.4	0.0124	1.6	1.5	1.5	0.0095	3.2	3.7	4.0
airfoil	0.0227	3.5	2.1	1.7	0.0208	4.3	3.7	2.1	0.0248	5.4	3.7	2.9
wine	0.0062	7.8	5.4	5.8	0.0120	2.4	1.4	1.6	0.0163	2.0	1.8	1.6
house16	0.0102	3.5	3.0	2.9	0.0158	1.6	1.2	1.4	0.0213	1.2	0.9	1.0

Table 12: **Grid refinement, the KAN as `pykan` trains it.** Both models refit with efficient-kan's `update_grid` called on the full train split ten times over the first 20 epochs (`pykan`'s schedule and its `batch=-1` default), then frozen; otherwise the primary protocol, five seeds. Each cell is fixed grid → refined. Refinement improves accuracy on most datasets and breaks training on `elevators`, `spambase` (either model's error more than doubles, marked ‡), where the comparison is moot; on the other 12 the additive<deep ordering holds on 12, with the median ratio at 2.0× against 4.9× on the fixed grid. Generated from `results/grid_update`.

	additive		deep KAN		
dataset	test	nRMSE	test	nRMSE	ratio
california	0.654→0.608	0.0057→0.0059	0.534→0.5	0.1406→0.0842	14.2×
airfoil	4.93→4.88	0.0183→0.0232	1.93→1.66	0.0908→0.0619	2.7×
wine	0.741→0.734	0.0075→0.0140	0.718→0.719	0.0509→0.0283	2.0×
cpu_act	2.93→2.79	0.0015→0.0147	2.42→2.38	0.0092→0.0169	1.1×
kin8nm	0.194→0.194	0.0036→0.0044	0.0732→0.0745	0.0093→0.0104	2.4×
elevators‡	0.00238→0.00639	0.0105→0.0200	0.00189→0.00615	0.0380→0.0087	0.4×
pumadyn	0.034→0.034	0.0036→0.0050	0.0221→0.0223	0.0070→0.0072	1.4×
wind	3.02→3.02	0.0090→0.0110	2.98→3	0.0242→0.0111	1.0×
house16	4.04e+04→3.93e+04	0.0084→0.0146	3.76e+04→3.53e+04	0.0407→0.0287	2.0×
phoneme	0.132→0.122	0.0122→0.0176	0.0772→0.0805	0.0698→0.0635	3.6×
magic	0.11→0.105	0.0131→0.0189	0.0786→0.0796	0.0473→0.0748	4.0×
jasmine	0.195→0.184	0.0031→0.0138	0.206→0.176	0.0176→0.0150	1.1×
sylvine	0.0189→0.0206	0.0121→0.0138	0.0188→0.0186	0.0248→0.0216	1.6×
spambase‡	0.0243→0.283	0.0019→0.0685	0.0237→0.277	0.0044→0.0519	0.8×

Table 13: **The hierarchy at a common Rashomon radius.** Retrainings of the two classes do not span the same ϵ: the deep KAN's held-out loss ranges 3.8–24.3$\times$ wider across seeds than the additive model's, on 14/14 datasets, and Proposition 1(i) bounds an additive model by $C\sqrt{\epsilon/\gamma}$ – so the measured ratio divided by $\sqrt{\epsilon_{\text{deep}}/\epsilon_{\text{add}}}$ has median 1.00, which is what a sceptic who thinks we merely measured optimization variance would predict. Here the deep model is not allowed that wider ball: $\epsilon^\star$ is the *additive* model's own full loss spread on that dataset, and only deep seed pairs whose held-out metrics differ by at most $\epsilon^\star$ are kept (the additive model keeps all of its pairs by construction). Ten seeds, 45 pairs per model. The ordering holds on 14/14 and the deep model's disagreement barely moves from its all-pairs value. The body instead quotes the *stricter* reading of the same radius, which is the one Proposition 1(i) bounds against: keep the seeds within $\epsilon^\star$ of each class's *own best* rather than of each other. That never inverts either (8/8, 11/11, 12/12 at $\epsilon^\star$, $2\epsilon^\star$, $3\epsilon^\star$) but is far thinner – only 8 of 14 datasets keep a deep pair at $\epsilon^\star$, the worst on 1. Generated from `results/matched_epsilon`.

	Rashomon radius		deep KAN nRMSE			
dataset	$\epsilon_{\text{deep}}/\epsilon_{\text{add}}$	pairs kept	all pairs	matched	additive	ratio
california	4.4$\times$	19/45	0.1421	0.1487	0.0070	21.3$\times$
airfoil	24.3$\times$	4/45	0.0761	0.0766	0.0218	3.5$\times$
wine	9.3$\times$	18/45	0.0456	0.0431	0.0064	6.7$\times$
cpu_act	5.0$\times$	21/45	0.0080	0.0068	0.0013	5.3$\times$
kin8nm	20.3$\times$	1/45	0.0095	0.0095	0.0032	3.0$\times$
elevators	3.8$\times$	17/45	0.0343	0.0386	0.0105	3.7$\times$
pumadyn	6.4$\times$	17/45	0.0064	0.0062	0.0037	1.7$\times$
wind	8.9$\times$	6/45	0.0219	0.0234	0.0079	3.0$\times$
house16	23.5$\times$	8/45	0.0344	0.0317	0.0099	3.2$\times$
phoneme	6.2$\times$	35/45	0.0668	0.0699	0.0273	2.6$\times$
magic	19.0$\times$	4/45	0.0472	0.0422	0.0186	2.3$\times$
jasmine	12.1$\times$	3/45	0.0156	0.0178	0.0034	5.3$\times$
sylvine	5.1$\times$	14/45	0.0293	0.0278	0.0113	2.5$\times$
spambase	9.0$\times$	11/45	0.0048	0.0043	0.0018	2.3$\times$

Table 14: **The rest of the registry.** The five binary datasets our primary suite does not use, run under the identical leakage-free protocol (held-out test $1-$AUC, five seeds, train-only scaling). The hierarchy holds on 4/5. The exception is `credit_g`, where the deep model does not beat the additive one at all (its accuracy gap is -0.5%, failing the 2% meaningfulness floor used throughout), so there is no advantage whose interpretability could be compared. `churn`, which inverted under an earlier within-sample protocol, holds here. Generated from `results/corrected_clf_excluded`.

	test $1-$AUC		mains nRMSE		
dataset	GAM	deepKAN	GAM	deepKAN	hierarchy
churn	0.2992	0.2300	0.0144	0.0181	holds
credit_g	0.4216	0.4236	0.0899	0.0777	**inverts**
eeg_eye	0.3746	0.2056	0.0162	0.1273	holds
electricity	0.1453	0.1273	0.0087	0.0790	holds
bank_market	0.1240	0.1230	0.0147	0.0580	holds

Table 15: **One explanation object for every class.** Each model read by the partial dependence of its full predictor on the same grid and background (additive and GrowKAN included; the primary tables read those two by their native terms). Regression: nRMSE on the target scale. Classification: nRMSE of the probability-scale PDP. *flips*: seed pairs × features where both retrainings give a 25th-to-75th-percentile effect above a threshold fixed before looking (5% of the target s.d.; 0.01 in probability) and the two signs disagree. The run reproduces every primary number to 10^{-5} except California (0.2% deep KAN, 0.7% GrowKAN nRMSE; GPU non-determinism). Five seeds. Generated from `results/audit_v2`.

		full-predictor PDP nRMSE				sign flips	
dataset	task	additive	GrowKAN	deep KAN	deep MLP	additive	deep KAN
california	reg	0.0057	0.0393	0.1408	0.0975	0/50	0/55
house16	reg	0.0084	0.0166	0.0407	0.0367	0/111	0/97
wind	reg	0.0090	0.0217	0.0242	0.0192	0/100	0/103
elevators	reg	0.0105	0.0164	0.0380	0.0439	0/100	0/120
cpu_act	reg	0.0015	0.0031	0.0092	0.0056	0/36	0/36
kin8nm	reg	0.0036	0.0049	0.0093	0.0094	0/80	0/80
pumadyn	reg	0.0036	0.0032	0.0070	0.0114	0/43	0/11
wine	reg	0.0075	0.0130	0.0509	0.0222	0/96	0/106
airfoil	reg	0.0183	0.0334	0.0908	0.0705	0/50	6/50
phoneme	clf	0.0020	0.0116	0.0207	0.0196	0/50	0/50
magic	clf	0.0039	0.0089	0.0169	0.0137	0/80	3/67
jasmine	clf	0.0009	0.0050	0.0067	0.0090	0/40	0/40
sylvine	clf	0.0030	0.0046	0.0082	0.0119	0/60	0/67
spambase	clf	0.0006	0.0010	0.0015	0.0021	0/50	0/56

Table 16: **Interaction discovery is reproducible but not identifiable.** Jaccard overlap of each screen's top-k pairs. *within*: the same screen refit on different seeds. *between*: different screens on the same data. *chance*: two uniform top-k draws from $\binom{d}{2}$, which is what makes a small overlap interpretable. Median within is 1.00 and median between 0.20; 44 of 140 between-pairs sit at or below chance. Screens sharing the binning *rule* and differing only in whose residual they read agree at 0.20, and screens using different rules at 0.20: the rule is not what they disagree about. Passing a cross-seed check is therefore necessary but not sufficient – a screen can score 1.00 against itself and still report an answer no other method recovers.

			within (seeds)					between (methods)									
dataset	d	chance	residual	gbm	linear	mlp	ebm	residual|gbm	residual|linear	residual|mlp	residual|ebm	gbm|linear	gbm|mlp	gbm|ebm	linear|mlp	linear|ebm	mlp|ebm
california	8	0.07	1.00	0.24	1.00	1.00	0.80	0.04	0.50	1.00	0.16	0.08	0.04	0.32	0.50	0.20	0.16
airfoil	5	0.20	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00	0.50	0.50	0.50	1.00	1.00	1.00
wine	11	0.03	1.00	1.00	1.00	0.54	0.65	0.00	0.50	0.44	0.00	0.00	0.00	0.12	0.44	0.00	0.00
cpu_act	21	0.01	0.70	1.00	1.00	0.65	0.60	0.00	0.00	0.16	0.00	0.00	0.08	0.00	0.20	0.20	0.44
kin8nm	8	0.07	1.00	1.00	1.00	0.16	1.00	0.00	1.00	0.08	1.00	0.00	0.08	0.00	0.08	1.00	0.08
elevators	18	0.01	1.00	1.00	1.00	0.00	0.70	0.00	0.00	0.04	0.50	0.00	0.00	0.00	0.14	0.00	0.04
pumadyn	32	0.00	1.00	1.00	1.00	0.20	0.60	0.20	0.50	0.20	0.20	0.20	0.20	0.20	0.26	0.20	0.20
wind	14	0.02	1.00	1.00	1.00	0.68	0.70	0.50	0.20	0.50	0.12	0.00	0.84	0.00	0.04	0.38	0.00
house16	16	0.02	0.80	1.00	1.00	1.00	1.00	0.00	0.90	0.44	0.00	0.00	0.00	0.00	0.50	0.00	0.00
phoneme	5	0.20	1.00	1.00	1.00	1.00	1.00	0.20	0.20	0.50	0.20	0.20	0.50	0.20	0.50	0.50	0.50
magic	10	0.04	1.00	1.00	1.00	0.70	1.00	0.20	0.50	0.20	0.20	0.20	0.00	0.50	0.00	0.20	0.00
jasmine	8	0.07	1.00	1.00	1.00	0.80	1.00	0.00	1.00	0.04	0.20	0.00	0.44	0.00	0.04	0.20	0.20
sylvine	20	0.01	0.70	1.00	1.00	0.44	1.00	0.20	0.70	0.44	0.50	0.20	0.22	0.20	0.60	0.50	0.60
spambase	57	0.00	1.00	1.00	1.00	1.00	0.47	0.50	0.20	0.20	0.00	0.20	0.50	0.00	0.50	0.00	0.00

Table 17: **Cost triple** (`california` d=8 and `pumadyn` d=32; measured separately from Fig. 6, which uses `airfoil`). FLOPs are reported two ways because the usual counter is misleading here: *matmul* counts only the matmul family, which for a KAN omits the entire B-spline recursion, while *MAC* counts a multiply–accumulate as one operation and each elementwise result as one; *FLOP* counts a multiply–accumulate as two, as it is. The matmul column shows how much of a KAN's work standard tooling misses; the two total columns show that whether that work makes a KAN more or less expensive than the MLP depends on which convention is used, so both are given. Generated from `results/phase4f_cost/cost.jsonl`.

Model	Params ↓	matmul only	total (MAC)	total (FLOP)	fwd+bwd (ms)
GAM (kan_shallow)	80	72	1,859	1,931	1.50
GROWKAN k=2 (kan)	320	288	4,771	5,059	9.03
GROWKAN k=2 (mult)	160	144	2,845	2,989	6.31
NAM	17,921	17,152	17,989	35,141	7.47
deep KAN [d,16,1]	1,440	1,296	6,680	7,976	6.04
deep MLP [d,64,64,1]	4,801	4,672	4,818	9,490	1.83
GAM (kan_shallow)	320	288	7,434	7,722	1.72
GROWKAN k=2 (kan)	560	504	10,346	10,850	9.22
GROWKAN k=2 (mult)	400	360	8,419	8,779	7.85
NAM	71,681	68,608	71,950	140,558	27.21
deep KAN [d,16,1]	5,280	4,752	15,518	20,270	3.83
deep MLP [d,64,64,1]	6,337	6,208	6,360	12,568	1.27

Table 18: **Knot-placement control.** Both models refit with a train-only rank transform so the spline knots see the data (*degen* = fraction of features with >90% of train mass in one knot cell). The additive<deep ordering survives on 14/14 under the rank transform (and 14/14 under min-max), so the hierarchy is not an artifact of rigid knots. What moves is the *margin*: the deep/additive ratio shrinks on 11/14 datasets, its median falls 4.2× to 2.0×, and the number of datasets where the ratio exceeds 1 by more than 2 jackknife SE drops from 12/14 to 7/14. The ordering is robust to knot placement; the effect *size* is only resolvable on about half the suite once knots see the data. Ratios carry leave-one-seed-out SEs; nRMSE lower = more identifiable.

	degen		additive nRMSE		deep nRMSE		ratio	
dataset	min-max	rank	min-max	rank	min-max	rank	min-max	rank
california	0.50	0.00	0.0057	0.0050	0.1407	0.0990	24.5±5.8	20.0±4.7
airfoil	0.00	0.00	0.0183	0.0349	0.0908	0.0601	4.95±1.69	1.72±0.78
wine	0.18	0.00	0.0075	0.0165	0.0509	0.0277	6.83±3.54	1.67±0.65
cpu_act	0.71	0.00	0.0015	0.0077	0.0092	0.0120	6.20±1.22	1.56±0.47
kin8nm	0.00	0.00	0.0036	0.0036	0.0093	0.0091	2.62±0.41	2.53±0.43
elevators	0.22	0.17	0.0105	0.0140	0.0380	0.0283	3.61±0.95	2.01±0.24
pumadyn	0.00	0.00	0.0036	0.0036	0.0070	0.0070	1.93±0.19	1.95±0.18
wind	0.00	0.00	0.0090	0.0022	0.0242	0.0207	2.67±0.30	9.26±1.46
house16	0.44	0.00	0.0084	0.0105	0.0407	0.0223	4.87±1.21	2.12±0.28
phoneme	0.00	0.00	0.0122	0.0094	0.0698	0.0566	5.72±3.71	6.05±0.92
magic	0.10	0.00	0.0131	0.0213	0.0473	0.0359	3.62±0.26	1.69±0.80
jasmine	0.62	0.00	0.0031	0.0076	0.0176	0.0162	5.63±0.96	2.12±0.58
sylvine	0.00	0.00	0.0121	0.0146	0.0248	0.0282	2.06±0.27	1.93±0.48
spambase	0.96	0.33	0.0019	0.0042	0.0044	0.0074	2.31±0.43	1.77±1.04

Table 19: **KAN's own sparsification does not close the gap.** Deep KAN $[d, 16, 1]$ retrained with `efficient-kan`'s L1+entropy regularizer at increasing weight, same leakage-free protocol. It helps where the PDP is worst (California $2.3\times$) and hurts where it is already good (`kin8nm`), never reaches the additive model, and costs the deep model its accuracy advantage.

dataset	reg. weight	test	PDP nRMSE
california	0	0.5335	0.14072
	0.01	0.6243	0.04979
	0.1	0.6597	0.06709
	1	0.6700	0.08074
	additive	0.6542	0.00574
airfoil	0	1.9278	0.09084
	0.01	2.3221	0.06160
	0.1	4.5857	0.10155
	1	5.3378	0.07188
	additive	4.9328	0.01834
kin8nm	0	0.0732	0.00933
	0.01	0.0819	0.01317
	0.1	0.1385	0.02449
	1	0.1419	0.01669
	additive	0.1943	0.00356

Table 20: **Leakage-free interaction-order diagnosis** (held-out test, train-only pair/triple screening; 5 seeds vary model init on one fixed split). Gap recovery is the % of the GAM→deepKAN test-RMSE gap closed by screened 2-way (+pair) vs. +3-way (+triple) blocks; Δ is the 3-way increment in points. A large Δ signals genuinely higher-order structure; $\Delta\approx 0$ means purely pairwise. Following the same pre-stated criterion as the corrected tables, percentages are reported only where the gap exceeds 2% of the GAM error (column *gap*); otherwise the ratio is unstable and we give the absolute gap alone. On a synthetic pure 3-way term (zero 1/2-way marginals, provably invisible to a pairwise screen), the top-1 screened triple matches the ground-truth triple with accuracy 1.00.

dataset	GAM	+pair	+triple	deepKAN	gap	pair rec%	triple rec%	Δpp
synth_3way	0.1948	0.1916	0.05584	0.06355	0.1312	2	106	+103
wine	0.7409	0.7362	0.7292	0.7183	0.02265	21	52	+31
kin8nm	0.1943	0.1804	0.1576	0.07317	0.1211	11	30	+19
california	0.6542	0.5929	0.5891	0.5335	0.1207	51	54	+3
elevators	0.002382	0.002251	0.002	0.001895	0.0004876	27	78	+51
pumadyn	0.03402	0.02143	0.02086	0.02214	0.01188	106	111	+5
pol	22.45	16.89	14.17	5.157	17.29	32	48	+16
fried	1.669	0.9946	0.9973	1.007	0.6623	102	101	-0
cpu_act	2.931	2.586	2.523	2.418	0.5131	67	79	+12

Table 21: **Multi-split robustness and significance.** Each dataset is re-evaluated under 5 independent fixed train/val/test splits (seeds 0–4). 'hier' counts splits where the additive $\text{kan}_{[d,1]}$ is more identifiable (lower cross-seed nRMSE) than the deep KAN; nRMSE and GrowKAN gap-recovery are mean±std over splits. Across datasets the additive<deep hierarchy holds on every split for all datasets; a one-sided binomial sign test over the per-dataset split-0 hierarchy gives p=0.00195 (regression 9/9), p=0.0312 (classification 5/5), p=$6.1e-05$ combined.

dataset	task	hier (/5)	GAM nRMSE	deepKAN nRMSE	rec%
california	regr	5/5	0.005±0.001	0.121±0.019	56±5
airfoil	regr	5/5	0.017±0.003	0.081±0.007	73±2
wine	regr	5/5	0.007±0.003	0.044±0.007	37±13
cpu_act	regr	5/5	0.002±0.000	0.009±0.001	72±4
kin8nm	regr	5/5	0.004±0.002	0.010±0.001	12±1
elevators	regr	5/5	0.010±0.001	0.031±0.006	25±6
pumadyn	regr	5/5	0.004±0.001	0.007±0.000	108±1
wind	regr	5/5	0.007±0.001	0.018±0.005	n/a
house16	regr	5/5	0.015±0.007	0.034±0.003	16±3
phoneme	clas	5/5	0.010±0.003	0.072±0.014	60±3
magic	clas	5/5	0.011±0.001	0.052±0.004	71±10
jasmine	clas	5/5	0.004±0.002	0.016±0.003	187±0
sylvine	clas	5/5	0.011±0.002	0.029±0.004	128±39
spambase	clas	5/5	0.002±0.000	0.006±0.001	125±45

Table 22: **Held-out test accuracy across 5 independent splits** (RMSE for regression, 1-AUC for classification; **lower is better**; mean±95% t-CI over splits, each split averaged over 5 init seeds). GrowKAN k=2 closes most of the GAM→deepKAN gap; CIs over splits are tight relative to the gap, so the recovery is not a split artifact.

dataset	task	GAM	deepKAN	GrowKAN k=2
california	regr	0.6686±0.0198	0.5432±0.0224	0.5984±0.0130
airfoil	regr	4.7373±0.2865	1.8794±0.0943	2.6418±0.0890
wine	regr	0.7298±0.0192	0.7045±0.0218	0.7181±0.0201
cpu_act	regr	2.9080±0.1957	2.3823±0.0758	2.5254±0.1011
kin8nm	regr	0.1958±0.0031	0.0728±0.0009	0.1815±0.0017
elevators	regr	0.0024±0.0001	0.0019±0.0001	0.0023±0.0001
pumadyn	regr	0.0332±0.0006	0.0221±0.0003	0.0212±0.0003
wind	regr	3.1366±0.0966	3.1008±0.0951	3.1273±0.0877
house16	regr	38440.8102±2709.2921	34987.8088±3011.6071	37886.9870±2790.6482
phoneme	clas	0.1190±0.0168	0.0694±0.0084	0.0894±0.0099
magic	clas	0.1053±0.0075	0.0770±0.0075	0.0853±0.0099
jasmine	clas	0.1833±0.0251	0.1839±0.0296	0.1780±0.0304
sylvine	clas	0.0278±0.0093	0.0248±0.0061	0.0226±0.0066
spambase	clas	0.0249±0.0047	0.0237±0.0039	0.0231±0.0052

Table 23: **The identifiability hierarchy is not a PDP artifact.** Deep-KAN shape disagreement (nRMSE) under partial dependence (PDP) vs. accumulated local effects (ALE, the standard less-biased alternative for correlated inputs), against the additive $\text{kan}_{[d,1]}$ (native extraction). The additive<deep hierarchy holds under *both* extractors (14/14 PDP, 14/14 ALE); ALE nRMSE is usually smaller than PDP – not on
textttkin8nm,
textttphoneme or
texttttspambase – but the gap to the additive model survives.

dataset	task	$\text{kan}_{[d,1]}$	deepKAN (PDP)	deepKAN (ALE)
california	regr	0.006	0.141	0.068
airfoil	regr	0.018	0.091	0.078
wine	regr	0.007	0.051	0.047
cpu_act	regr	0.001	0.009	0.006
kin8nm	regr	0.004	0.009	0.010
elevators	regr	0.011	0.038	0.024
pumadyn	regr	0.004	0.007	0.007
wind	regr	0.009	0.024	0.019
house16	regr	0.008	0.041	0.034
phoneme	clas	0.012	0.070	0.081
magic	clas	0.013	0.047	0.035
jasmine	clas	0.003	0.018	0.017
sylvine	clas	0.012	0.025	0.024
spambase	clas	0.002	0.004	0.007

Table 24: **The hierarchy is not an artifact of comparing unequally-accurate retrainings (C1).** Restricting cross-seed disagreement to *predictive-equivalent* seed pairs—both seeds within 10% of that model's best held-out metric—the additive $\text{kan}_{[d,1]}$ keeps all $10/10$ pairs everywhere (its accuracy barely moves), while the deep KAN sometimes retains as few as 1–3; among those equally-good pairs the additive<deep gap persists ($9/9$ regression, $5/5$ classification). 'all' = every pair (sensitivity). nRMSE, lower=more identifiable.

	eligible pairs (/10)		deepKAN nRMSE	
dataset	GAM	deepKAN	equal-acc	all-pair
california	10	10	0.141	0.141
airfoil	10	3	0.055	0.091
wine	10	10	0.051	0.051
cpu_act	10	10	0.009	0.009
kin8nm	10	10	0.009	0.009
elevators	10	10	0.038	0.038
pumadyn	10	10	0.007	0.007
wind	10	10	0.024	0.024
house16	10	10	0.041	0.041
phoneme	10	1	0.067	0.070
magic	10	10	0.047	0.047
jasmine	10	10	0.018	0.018
sylvine	10	3	0.024	0.025
spambase	10	10	0.004	0.004

Table 25: **Science axis (Feynman), an honest boundary.** On symbolic-regression equations—KANs' advertised strength—additivity is the *wrong* hypothesis class: the additive $\text{kan}_{[d,1]}$ badly underfits (large GAM→deepKAN test-RMSE gaps) and, tellingly, is *no longer the most identifiable* model (the hierarchy inverts on 10/10 equations—an additive model that misfits gives unstable shapes). But this is exactly where GrowKAN's screened, purified interaction blocks matter most: k=2 blocks recover a median 67% of the gap (up to 99% on two-variable equations). RMSE and cross-seed nRMSE, lower=better.

		test RMSE				nRMSE		
equation	d	GAM	deepKAN	GrowKAN	rec%	GAM	PDP	ALE
I.11.19	6	0.572	0.012	0.338	42	0.000	0.000	0.000
I.12.2	3	0.060	0.001	0.021	65	0.010	0.006	0.006
I.13.12	4	0.442	0.011	0.317	29	0.002	0.001	0.001
I.15.3t	2	0.358	0.006	0.013	98	0.001	0.000	0.001
I.26.2	2	0.185	0.001	0.002	99	0.018	0.002	0.002
I.29.16	4	0.601	0.028	0.226	65	0.015	0.005	0.006
I.6.2	2	0.077	0.001	0.002	99	0.033	0.015	0.016
II.11.28	2	0.023	0.000	0.000	99	0.130	0.009	0.009
II.2.42	4	0.639	0.009	0.427	34	0.003	0.000	0.000
III.15.12	3	2.243	0.021	0.722	68	0.007	0.000	0.000

Table 26: **Validating GrowKAN's interaction discovery.** Left: on synthetic data with *known ground-truth* pairs, the residual screen's top-k recovers them (recall@k / precision), including under concurvity (ρ)—an empirical recovery result in these settings. Right: on real data (no ground truth) the screened pairs are compared to the pairwise interactions an independent detector (EBM-GA2M) auto-selects as most important (top-3 overlap): convergent validity, full agreement where interaction structure is strong (kin8nm, airfoil) and none on two datasets (cpu_act, wine).

synthetic (ground truth)

ρ	#pairs	recall@k	precision
0.0	1	1.00	1.00
0.0	2	1.00	1.00
0.0	3	1.00	1.00
0.5	1	1.00	1.00
0.5	2	0.90	0.90
0.5	3	0.90	0.90

real (agreement with EBM-GA2M top-3)

dataset	overlap	Jaccard
california	1/3	0.20
cpu_act	0/3	0.00
kin8nm	3/3	1.00
pumadyn	1/3	0.20
wine	0/3	0.00
airfoil	3/3	1.00

Table 27: **Additive identifiability is basis-independent in ordering.** Cross-seed shape nRMSE (lower=more identifiable) for the additive $[d, 1]$ model vs. a non-additive model with the *same* univariate basis (B-spline / Gaussian RBF / Chebyshev; PDP for the latter). The additive<deep ordering holds in 15/15 dataset×basis cells (clearly on 12; the `cpu_act` cells are near-ties where both are already near-perfectly identifiable). Magnitude is basis-dependent (RBF/Cheb additive is $\sim 2\times$ less identifiable than spline) but the ordering is not. The deep RBF/Cheb models place the basis at the input and share a plain MLP mixer (not per-edge basis KANs).

	B-spline		RBF		Chebyshev	
dataset	add	deep	add	deep	add	deep
california	0.006	0.133	0.003	0.042	0.007	0.048
cpu_act	0.000	0.000	0.000	0.000	0.001	0.001
kin8nm	0.012	0.035	0.047	0.075	0.036	0.147
airfoil	0.003	0.012	0.003	0.015	0.004	0.016
wine	0.007	0.055	0.014	0.030	0.019	0.030

Table 28: **Interaction screening matters.** GrowKAN k=2 gap recovery (% of GAM→deepKAN RMSE gap) when the two pairs are chosen by residual interaction screening vs. at random. Screened pairs recover substantially more of the gap, so the gain is from *which* interactions are grown, not merely from added block capacity.

dataset	GAM	deepKAN	screened rec%	random rec%
california	0.6542	0.5335	51	25
cpu_act	2.9307	2.4176	67	5
kin8nm	0.1943	0.0732	11	2

Table 29: **Identifiability is robust to spline grid density.** Additive $\text{kan}_{[d,1]}$ cross-seed shape disagreement (nRMSE) at grid resolutions 32/64/128; the additive shapes stay identifiable across densities.

dataset	grid 32	grid 64	grid 128
california	0.006	0.005	0.005
cpu_act	0.000	0.000	0.000
kin8nm	0.014	0.013	0.013

Table 30: *Within-sample diagnostic (secondary to the leakage-free tables).* Regression cross-seed shape disagreement (nRMSE, lower=more identifiable). Additive models (first three) vs deep partial-dependence (last two).

dataset	$\text{kan}_{[d,1]}$	EBM	NAM	deepKAN	deepMLP
california	0.010	0.023	0.064	0.280	0.752
airfoil	0.043	0.032	0.064	0.077	0.163
wine	0.023	0.030	0.046	0.064	0.039
wind	0.014	0.017	0.019	0.029	0.028
kin8nm	0.013	0.012	0.016	0.011	0.016
cpu_act	0.006	0.047	0.015	0.024	0.038
pumadyn	0.028	0.015	0.028	0.026	0.025
elevators	0.014	0.024	0.037	0.134	0.141
house16	0.032	0.029	0.062	0.045	0.090

Table 31: *Within-sample diagnostic (secondary to the leakage-free tables).* Scaled 10-seed identifiability sweep over 14 regression datasets, including high-dimensional (`superconduct` 81d, `pol` 48d, `ailerons` 40d) and larger (n up to 20k). Additive<deep hierarchy holds on 12/14, but the margin varies widely: 21× on `california` down to 1.0× on `fried`. nRMSE, lower=more identifiable.

dataset	$\text{kan}_{[d,1]}$	EBM	NAM	deepKAN	deepMLP
california	0.015	0.023	0.063	0.316	0.640
airfoil	0.055	0.034	0.067	0.069	0.155
wine	0.021	0.028	0.045	0.067	0.040
wind	0.015	0.016	0.018	0.032	0.031
kin8nm	0.015	0.012	0.016	0.011	0.016
cpu_act	0.009	0.044	0.014	0.018	0.031
pumadyn	0.033	0.014	0.027	0.023	0.034
elevators	0.016	0.020	0.051	0.107	0.113
house16	0.026	0.028	0.040	0.057	0.058
pol	0.017	0.017	0.058	0.035	0.023
ailerons	0.010	0.012	0.017	0.018	0.025
superconduct	0.043	0.029	0.063	0.050	0.054
diamonds	0.073	0.050	0.183	0.122	0.114
fried	0.006	0.008	0.012	0.006	0.011

Table 32: *Within-sample diagnostic (secondary to the leakage-free tables).* Classification (1-AUC, lower=better) and logit-scale identifiability (nRMSE). 'rec%' = GrowKAN k=2 gap recovery (n/a when GAM≈deep).

	accuracy (1-AUC)			identifiability (nRMSE)		
dataset	GAM	deepKAN	rec%	$\text{kan}_{[d,1]}$	deepKAN	deepMLP
credit_g	0.358	0.354	n/a	0.217	0.214	0.128
phoneme	0.128	0.063	56	0.130	0.150	0.135
magic	0.108	0.073	66	0.033	0.081	0.110
electricity	0.146	0.121	84	0.042	0.112	0.021
bank_market	0.126	0.120	n/a	0.032	0.101	0.074
jasmine	0.187	0.172	78	0.038	0.136	0.059
sylvine	0.029	0.021	n/a	0.041	0.068	0.417
churn	0.292	0.222	42	0.055	0.038	0.038
spambase	0.024	0.022	n/a	0.011	0.030	0.045
eeg_eye	0.387	0.302	3	0.055	0.403	0.034

Table 33: *Within-sample diagnostic (secondary to the leakage-free tables).* GrowKAN k=2 gap recovery (% of GAM→deepKAN RMSE gap) and main-shape nRMSE, by block type, on the regression suite.

	GAM	deepKAN	kan block		mult block	
dataset	RMSE	RMSE	rec%	nRMSE	rec%	nRMSE
california	0.668	0.534	57	0.057	4	0.019
airfoil	4.707	1.704	73	0.048	52	0.063
wine	0.728	0.693	46	0.046	44	0.042
wind	3.128	3.067	29	0.032	17	0.043
kin8nm	0.195	0.070	11	0.014	11	0.018
cpu_act	2.861	2.323	72	0.180	66	0.044
pumadyn	0.033	0.022	105	0.092	105	0.055
elevators	0.002	0.002	24	0.128	23	0.091
house16	38424.659	34005.269	31	0.058	4	0.062

Table 34: **Leakage-free head-to-head** with interaction-GAM competitors (held-out test RMSE for regression, 1-AUC for classification; **lower is better**; train-only scaling, same fixed split as Table 1; up to 5 seeds each, divergent NodeGAM seeds skipped). GrowKAN uses only $k{=}2$ pair blocks regularized toward purity ($\sim$300 params); on the held-out test it has lower error than NodeGAM (neural GA2M) on 6/14 tasks at a median $69\times$ fewer parameters, while EBM-GA2M (boosted-tree GA2M, many auto-selected interactions) is the accuracy ceiling. The NodeGAM column is its library default; we also tuned it over a validation-selected grid on 14/14 datasets, which improved it on 11, and the count above uses whichever of the two is lower. Generated from `results/corrected`.

dataset	task	GrowKAN $k{=}2$	GK params	EBM-GA2M	NodeGAM	NG params
california	regr	0.5924	320	0.4445	0.5887	19860
airfoil	regr	2.7385	290	2.1864	3.4255	16260
wine	regr	0.7362	350	0.6855	0.7225	23460
cpu_act	regr	2.5860	450	2.3540	2.6957	35460
kin8nm	regr	0.1804	320	0.1620	0.1825	19860
elevators	regr	0.0023	420	0.0022	0.0027	31860
pumadyn	regr	0.0214	560	0.0236	0.0225	48660
wind	regr	3.0100	380	2.9801	3.0216	27060
house16	regr	39961.9765	400	34657.6827	37278.7832	29460
phoneme	clas	0.1003	290	0.0716	0.1067	16260
magic	clas	0.0934	340	0.0782	0.0948	22260
jasmine	clas	0.1874	320	0.1842	0.1826	19860
sylvine	clas	0.0177	440	0.0128	0.0185	34260
spambase	clas	0.0241	810	0.0131	0.0209	78660

Table 35: *Within-sample diagnostic (secondary to the leakage-free tables).* Interaction-order diagnosis across 9 datasets: gap recovery (% of GAM$\rightarrow$deepKAN gap) by 2-way vs. +3-way blocks. A large +3-way jump signals higher-order structure; no jump means purely pairwise. `fried` (Friedman, ground-truth 2-way only) is correctly diagnosed pairwise; `synth_3way` (pure triple) is invisible to 2-way (-0%) and fully recovered by 3-way. Superseded by the leakage-free Table 20; retained as a cross-check.

dataset	GAM	+pair	+triple	deepKAN	pair rec%	triple rec%
synth_3way	0.195	0.195	0.052	0.054	-0	102
synth_2way	0.477	0.052	0.052	0.053	100	100
kin8nm	0.196	0.181	0.160	0.070	12	29
california	0.654	0.575	0.552	0.513	56	72
wine	0.734	0.716	0.717	0.701	54	53
cpu_act	2.830	2.472	2.440	2.347	74	81
elevators	0.002	0.002	0.002	0.002	22	67
pol	22.324	16.640	16.412	4.471	32	33
fried	1.660	0.989	0.990	1.002	102	102

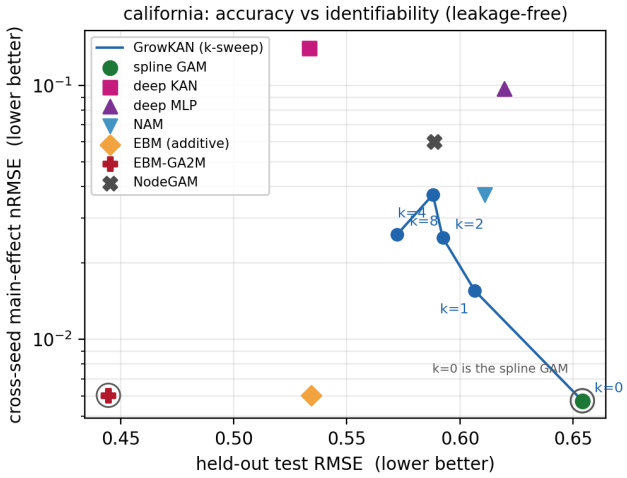

Figure 7: **The same k-sweep on California, where the additive control says the gap is *not* about interactions** (leakage-free, same protocol as Fig. 6). Circled points are Pareto-optimal. Here the outcome inverts: every $k \geq 1$ point is dominated, and the dominating model is the boosted GA2M (red), more accurate than any GROWKAN grown on this dataset without giving up any main-effect stability to get there. The two EBMs make that concrete—they sit at *exactly* the same height (both $\mathrm{nRMSE} = 0.006031$, agreeing to every digit we store) and differ only along the accuracy axis, which is the bit-identical main-effect property of §9 rendered as a picture: the pairwise terms bought accuracy without moving EBM's *native* main effects, so the additive EBM is itself dominated. Read by PDP of the full predictor, which folds the interactions in, EBM-GA2M is $2.3\times$ looser on California (App. C). Only $k{=}0$, the spline GAM, survives alongside it, and only because no model here disagrees with itself less. Fig. 6 reaches the same verdict on `airfoil`, the case most favourable to growing interactions; that GROWKAN is dominated on both is what the $3/14$ of §9 looks like. The contrast between the two panels is not who wins but *why* growing is futile here: the additive control attributes 1% of California's gap to interactions against 97% of `airfoil`'s, so this is the negative case the control identifies in advance.