IN-CONTEXT LEARNING AS QUERY-CONDITIONED HYPOTHESIS SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

In-Context Learning (ICL) is the ability of Large Language Models (LLMs) to alter their input–output mapping by conditioning on demonstrations in the prompt, without parameter updates. Beyond operational differences, ICL is different from Empirical Risk Minimization in its generalization behavior, calling for new explanatory frameworks. Theoretical accounts explain ICL as different forms of adaptation to the demonstrations, ranging from Bayesian inference over hypotheses with demonstrations as evidence to implicit fitting of a learned predictor to the demonstrations. We model ICL as case-based, transductive learning in which LLM predictions reflect query-conditioned switching among predictive modes. We formulate this view and test it in controlled regression tasks by analyzing prediction surfaces for branch structure, measuring query-dependent shifts in probability mass of prediction, and testing if causal interventions on model activations can selectively induce or suppress predictive modes. This perspective reconciles disparate theories of ICL by interpreting each implicit learner accounts as descriptions of behavior in different query–demonstration regimes.

1 INTRODUCTION

The input–output mapping of Large Language Models (LLMs) can be modified at inference time by conditioning on tokens in context, without any parameter updates. This enables LLMs to learn new tasks at inference by modifying context to include task-specific instructions (Wei et al., 2021), demonstrations (Brown et al., 2020), or explicit reasoning traces (Wei et al., 2022). Demonstration-based In-Context Learning, in particular, has been central to the widespread use of LLMs (Dong et al., 2024), enabling few-shot task learning in LLM-based assistants and environment inference from a small number of observations in LLM-based agents.

For the same set of data points, In-Context Learning(ICL) exhibits a distinct form of learning compared to Empirical Risk Minimization (ERM), both in its operational mechanism and generalization. On compositional tests, ICL is often more data efficient and generalizes better than ERM style fine-tuning approaches, which typically require richer semantic supervision or targeted augmentations (Berglund et al., 2024; Lampinen et al., 2025). ICL can be more robust than fine-tuning under distribution shift(Awadalla et al., 2022), and contexts can override ERM-trained behaviors when the two disagree (Anil et al., 2024). These observations suggest that understanding ICL requires new theoretical explanations beyond traditional ERM based learning theories.

Existing theories mostly explain ICL as different forms of adaptation to the demonstrations. Algorithmic accounts model the forward pass as fitting a predictor to the demonstrations, behaviorally resembling classical estimation or gradient descent. (Garg et al., 2022; Akyürek et al., 2023; Von Oswald et al., 2023). Bayesian accounts instead describe ICL as inference over latent tasks acquired during pretraining (Xie et al., 2022; Raventós et al., 2023), while kernel accounts describe prediction through similarity-weighted interpolation of the demonstrations (Han et al., 2023). These perspectives explain important aspects of ICL, yet their success depends on the regime: models depart from the predictions of fixed learning algorithms (Shen et al., 2024; Deutch et al., 2024) and violate consistency properties required by Bayesian inference (Falck et al., 2024; Kossen et al., 2024). Recent work therefore models ICL as a mixture of learning strategies, examining how the demonstrations (Bai et al., 2023), the training regime (Park et al., 2025; Wurgaft et al., 2025), and the task structure (Hosseini et al., 2026) determine which strategy is expressed.

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.

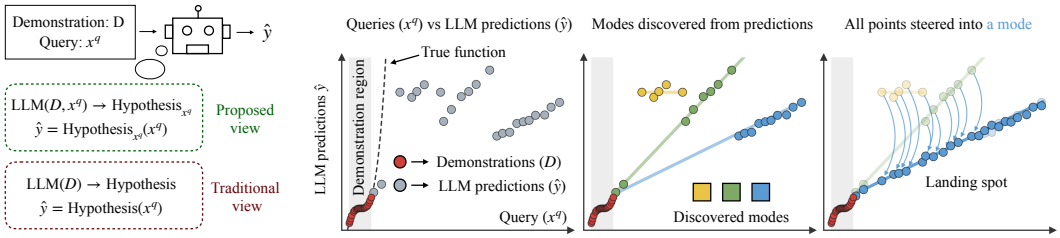

Figure 1: **ICL is query conditioned hypothesis selection.***Left:* The traditional view explain ICL as deriving one hypothesis (or one mixture) from the demonstrations D. We propose that at each query Q, Q also selects which hypothesis is used. *Right: Gemma-3-12B*, demonstrations sampled from a cubic function, with few LLM predictions sampled for clarity. Beyond the demonstrations, LLM seems to use different hypotheses for prediction, which we discover as *modes*, steering with one mode's direction pulls all queries into it.

We go beyond the view that ICL is solely an adaptation to the demonstrations by modeling it as query-conditioned selection of *predictive modes*: candidate hypotheses about the input–output relation derived from the demonstrations. We model ICL as *case-based, transductive* learning, where the demonstrations induce a repertoire of *predictive modes*, and the query determines which of them is expressed (Figure 1, left). Interestingly, this view is analogous to the long established theory in human function-learning research, where the hypothesis a learner expresses depends jointly on the training examples and the query (DeLosh et al., 1997; Kalish et al., 2004b; Lucas et al., 2015).

We turn this analogy into a falsifiable theory and test it on controlled regression tasks similar to those used in prior work (Garg et al., 2022; Akyürek et al., 2023; Xie et al., 2022), against two central alternatives: a fixed learner with growing noise, and a fixed mixture over a hypothesis family whose weights depend only on the demonstrations. We investigate these alternatives on the transductive prediction surface: output trace of LLM generated by holding demonstrations fixed and varying the query. Peeling this output trace reveals coherent branches, with excess fragmentation relative to the same decomposition of the ground-truth functions showing the multiple modes in transductive prediction. We then show activation steering to give causal evidence to the existence of the peeled modes. The analysis shows that a specific dominant mode discovered by peeling the prediction surface can be selectively induced on a held out query by just suppressing current mode expressed at that query.

We then use pointwise attribution to examine how predictive behavior changes with the query. Independent of peeling, we compare each prediction with those of classical predictors fitted to the demonstrations (using these predictors as behavioral proxies for known inductive biases). These comparisons reveal systematic shifts in the best-matching predictive families across query-demonstration regimes. However, a changing best match does not establish that the model's probability shifts between hypotheses. We therefore examine the probabilities assigned to numerical answers, using candidate hypotheses defined before evaluation. With demonstrations held fixed, we find that the query redistributes output probability in ways incompatible with purely demonstration based explanations. This shows further evidence of our proposal: the selection for predictive modes depends on the query as well as the demonstrations.

The proposed lens connects LLMs' jagged prediction profiles to the expression of different, sometimes inconsistent, predictive modes across queries. In short, we make the following contributions:

- A behavioral theory for in-context learning as case-based, transductance, where demonstrations induce candidate predictive modes, and the query influences which mode is expressed.
- Provide evidence for multiple predictive modes and query-dependent shifts in their probability support while holding demonstrations fixed. We test this account against a fixed learner and a fixed mixture of hypothesis whose weights depend only on the demonstrations.
- We characterize predictive behavior through complementary analyses: pointwise attribution tracks which hypotheses best match individual predictions, peeling reveals coherent branches in prediction surfaces, and causal interventions selectively induce or suppress the corresponding behaviors on held-out queries.

2 RELATED WORK

Since GPT-3 came out and its striking ability to learn from a few demonstrations was benchmarked (Brown et al., 2020), ICL has gained widespread attention from researchers (surveys: (Zhao et al., 2023; Minaee et al., 2024; Zhou et al., 2024; Luo et al., 2024)). We take the prominant theorotical accounts and bin them into three major categories (a) Implicit learning algorithms, (b) Bayesian-inference (c) Kernel and similarity-based accounts.

Implicit learning algorithms and meta-learning. Theoretical constructions show that transformers can implement classical estimators and optimization procedures, including least-squares regression, ridge regression, and gradient-based updates (Bai et al., 2023; Akyürek et al., 2023; Von Oswald et al., 2023). Complementing these constructions, Zhang et al. (2024) analyze gradient-flow training of a single linear self-attention layer and establish prediction guarantees for new linear-regression tasks. However, subsequent work questions extending these explanations to pretrained language models, the generality of the gradient-descent correspondence (Shen et al., 2024; Deutch et al., 2024). Our analysis builds on studies of prediction functions under fixed demonstrations (Garg et al., 2022), examining whether these surfaces exhibit distinct predictive modes.

Bayesian inference over latent hypotheses. Bayesian accounts describe ICL as inference over latent tasks or data-generating processes acquired during pretraining. Xie et al. (2022) develop this account for sequence distributions with shared latent concepts and investigate it using a synthetic language dataset generated from hidden Markov models. In regression, Raventós et al. (2023) find that increasing pretraining task diversity produces a transition from behavior resembling the Bayesian estimator under the finite pretraining task distribution to ridge regression. The latter is Bayesian under a Gaussian task prior, but departs from the Bayesian estimator associated with the actual pretraining distribution. Tests on pretrained LLMs also reveal violations of martingale conditions and deviations from Bayesian uncertainty scaling (Falck et al., 2024). Related behavioral studies find that ICL can learn new label relationships while retaining pretraining preferences and preferentially using information later in the context (Kossen et al., 2024). Our tests address a specific distinction within hypothesis-based accounts: whether predictions across queries remain compatible with a mixture over a specified hypothesis family whose weights depend only on the demonstrations. This comparison concerns the source of hypothesis reweighting, rather than Bayesian inference in general.

Kernel, retrieval, and similarity-based accounts. Similarity-based accounts surprisingly explain a large part of LLM prediction through the influence of demonstrations relevant to the query. Han et al. (2023) develop a kernel-regression account of ICL and examine its correspondence with language-model attention and hidden representations. Related work shows that selecting semantically similar demonstrations can improve performance (Liu et al., 2022), and that effective demonstration retrieval can be learned from feedback (Rubin et al., 2022). These accounts indirectly assign the query an explicit role in determining which examples influence prediction. Our focus is on whether the query also changes which predictive hypothesis is expressed. We therefore distinguish variation in the influence of demonstrations within a predictor from variation in support across candidate predictors.

Towards a unifying framework. Recent work increasingly explains ICL through multiple learning strategies, making the conditions governing their expression central to understanding model behavior. Transformers can select among algorithms using in-context evidence (Bai et al., 2023), while context length and training conditions can shift the balance between competing strategies (Park et al., 2025). Wurgaft et al. (2025) explain such shifts through a Bayesian account of how pretraining shapes preferences for memorizing and generalizing predictors. Converging evidence comes from Hosseini et al. (2026), who find that representational changes from ICL come with some tasks but not others, motivating a view of LLMs as selecting from a toolkit of strategies according to task structure. Together, these accounts support the view that ICL can express different predictive strategies depending on the setting. The remaining question we address is whether the query changes which mode is expressed, even when the model, task, and demonstrations is fixed.

Algorithmic accounts explain LLM using known inductive biases on the demonstrations, Bayesian accounts describe hypothesis reweighting with demonstration as evidence, and similarity-based accounts emphasize locality. Our case-based, transductive view examines when these descriptions apply: the same demonstrations can elicit different predictive behaviors across queries. This perspective draws on human function-learning accounts in which learners express different locally valid hypotheses depending on the test input, as discussed in Appendix A.

3 METHOD

This section formalizes the proposed view and outline the framework with which we test it. We first describe a case-based prompting framework, then the mode discovery method and the evaluation strategy for changing modes with queries. We also specify further test on model output probability and activations to validate the proposed view beyond output attribution and analysis.

Transductive predictive surface: Similar to prior work, we analyze ICL through controlled regression tasks. Let G be the function that generate data points $D = \{(x_i^D, y_i^D)\}_{i=1}^N$, such that $y_i^D = G(x_i^D)$. Every prompt contains n demonstration points together with a single query input x^q, formatted with an instruction to predict its output:

$$(x_1^D \mapsto y_1^D), \ldots, (x_n^D \mapsto y_n^D), \qquad x^q \mapsto ?$$

Given this prompt, the LLM returns a completion from which we extract the numerical prediction $\hat{y}^q$. The exact prompt template (similar to Vacareanu et al. (2024)) and the output extraction strategy is given in Appendix B.

We construct a fresh prompt for each query, keeping the model, prompt template, demonstrations, and their order fixed. Collecting the greedy predictions at m test queries gives $\{(x_j^q, \hat{y}_j)\}_{j=1}^m$ which are m points from the transductive prediction surface $F_{\mathrm{ICL}}(\cdot \mid D)$. $F_{\mathrm{ICL}}(\cdot \mid D)$ characterized by the m points reveals changes in predictive behavior under fixed demonstrations and changing query. Importantly, the branch structure in $F_{\mathrm{ICL}}(\cdot \mid D)$ dismisses the single mode prediction hypothesis with growing noise. We evaluate $F_{\mathrm{ICL}}(\cdot \mid D)$ over range containing both interpolation (in-support of demonstration) and extrapolation (out-of-support of demonstration). These terms are not to be confused with model's pretraining distribution, which is generally unknown.

Mode discovery: To discover branching structures in the geometry of $F_{\mathrm{ICL}}(\cdot \mid D)$, we treat $\{(x_j^q, \hat{y}_j)\}_{j=1}^m$ as data and apply outlier robust classic peeling algorithm on it: RANSAC (Fischler & Bolles, 1981). Peeling repeatedly fits a deliberately narrow class of functions (line or low-degree polynomial) on the points by maximizing agreement within a fixed tolerance, then removes the matched predictions, and continues on the remaining points. Applying the same procedure to G provides a baseline for fragmentation introduced by the fitting function class. Several peeled branches can still approximate a single curve. We therefore also examine small ranges of queries, comparing two contiguous line segments with two branches that alternate across nearby inputs. Controls with growing noise test whether these local comparisons distinguish branch structure from increasingly variable predictions.

These analyses find modes from the prediction surface post hoc, but do not establish how it relates to the model's internal representations. We therefore construct steering directions by contrasting activations on prompts expressing different modes, and add or subtract these directions on held-out prompts. Positive injection tests whether the target mode can be induced on a query where it was not originally expressed and negative injection tests what happens to the prediction when a mode is suppressed. If the output on which queries land when their true mode is suppressed is already explained by another mode obtained by the peeling of $F_{\mathrm{ICL}}(\cdot \mid D)$, its strong evidence that the modes correspond to the hypothesis used by the LLM.

Query changing predictive modes: Peeling describes the geometric form of a mode on the prediction surface. Independent of peeling, pointwise attribution compares each LLM prediction $\hat{y}$ with K different classical predictors fitted to the demonstrations. These predictors serve as behavioral proxies: their best matches describe which predictive biases the LLM resembles at each query, without assuming it internally implements those algorithms. We make the matching sensitive to the level of separation with similar matches treated as ambiguous. We compare the output of the K classical predictors fit on demonstration $c_k^D(x_i)$ with $\hat{y}_i$, especially for x_i far from the demonstrations as the overlap of predictors near demonstration is high.

However, a changing best match does not establish that the model's probability shifts between hypotheses. We therefore examine the probabilities assigned to numerical answers. For each candidate, we measure probability in a numerical interval around its prediction, conditional on a numerical answer. We obtain probability bounds by expanding possible answer strings using the model's token probabilities, retaining uncertainty for strings not reached by the computation. A fixed-mixture account assigns a probability weight to each hypothesis using the demonstrations alone.

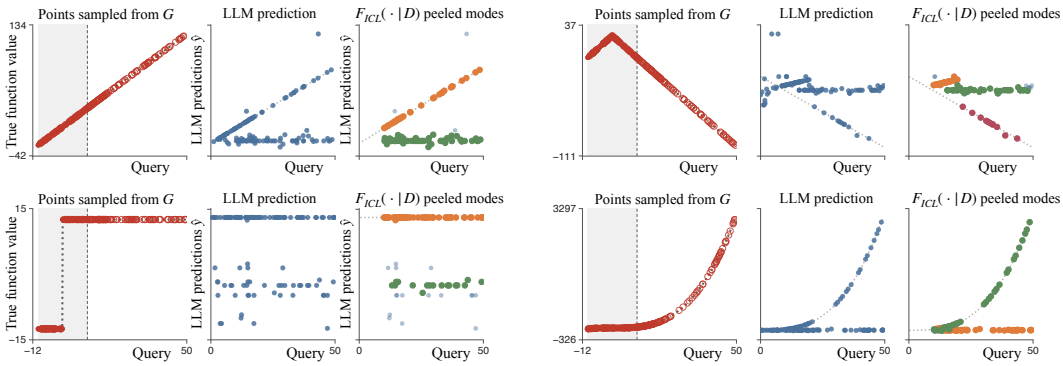

Figure 2: **Peeled decomposition of $F_{\mathrm{ICL}}(\cdot|D)$ reveals coexisting predictive modes within a single prompt.** Four representative $\mathcal{G}$ evaluated on *GPT-OSS* fixed to 3.6B at $n_{\mathrm{demos}} = 100$, in range $[-10, 10]$ (shaded) test queries in $x \in [0, 50]$. Each cell has three panels. *Left:* the true function $\mathcal{G}$ values at demonstrations and the test queries range. *Middle:* the LLM's raw predictions on the same test queries (blue). *Right:* the same pool re-colored by its RANSAC-traced mode. $\mathcal{G}$ are arranged row-wise, left to right. **Top row:** `linear`, `piecewise-linear`. **Bottom row:** `step`, `cubic`.

Relating these weights to the measured probabilities requires a bound on how much probability each hypothesis places outside its interval. Under this assumption, we test whether one set of weights can explain the measurements across queries, accounting for overlapping intervals, answers outside the family, and unresolved probability.

Finally, the unknown pretraining distribution limits how we interpret extrapolation in pretrained LLMs. We therefore also study models with known training distributions, separating queries beyond the demonstration range from those beyond the training range. This help scope the proposed view along with other limitations M.

4 Experiments and Results

We empirically evaluate the proposed framework across a diverse set of LLMs and controlled regression tasks. Our experiments are designed to isolate *behavioral* properties of in-context learning under fixed contexts rather than optimize task performance, and are organized so that each experiment targets one falsifiable claim of the query-conditioned-mode framework.

models: We evaluate eight autoregressive LLMs spanning four model families and parameter scales from 3B to 14B: *Llama-3.2-3B*, *Llama-3.1-8B*, *Qwen2.5-7B*, *Qwen2.5-14B*, *Gemma-3-4B*, *Gemma-3-12B*, *DeepSeek-R1-Distill-Qwen-14B*, and *GPT-OSS*. All models are queried in a zero-shot, zero-parameter-update setting with identical prompt templates (Appendix B), keeping the hardware mostly comparable (compute details in Appendix H).

To check whether query-conditioned predictive modes arise, we have to look at dense models without explicit expert routing. Several recent large language models use sparse mixtures of experts, including (Gemini Team, 2024; DeepSeek-AI, 2024). In these architectures, routers select a small subset of experts for each token at each MoE layer (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022), and routing analyses reveal systematic associations with token and input characteristics (Jiang et al., 2024; Muennighoff et al., 2025). Hence a changing hypothesis can explain the changes in predictive behavior, giving a simpler explanation to any observed mode switching. Dense models exclude this explicit routing mechanism, so they form the basis of our main conclusions; GPT-OSS-20B is retained as a fixed model with 3.6B parameters(OpenAI, 2025).

Query sweeps: To compare queries at similar degrees of extrapolation across tasks, we measure their distance from the demonstration input range relative to its width. Let $x^D_{\min}$ and $x^D_{\max}$ be the smallest and largest inputs among the demonstrations in in a prompt. We define the normalized extrapolation distance as

$$\delta(x^q) = \frac{\mathrm{dist}(x^q, [x^D_{\min}, x^D_{\max}])}{x^D_{\max} - x^D_{\min}}.$$

Table 1: The table shows the fragmentation of G/ fragmentation under peeling of $F_{ICL}(\cdot|D)$ under the same *cfg* averaged over eight LLMs. Cells colored green are over-fragmentation of the prediction surface; red for over-fragmentation of $\mathcal{G}$ relative to $F_{ICL}(\cdot|D)$, i.e. mode collapse. The first three rows are line-only attribution at inlier bands 1.5%, 3%, 6%, each averaged over 5 RANSAC seeds. *cfgs* is the mean over the full 30-cfg sweeps.

cfg	affine	cubic	dsin	exp	log	p-linear	quad	saw	sin	step
1.5%	1/3.5	4/3.9	4.6/4.6	1/4.2	3/2.4	2/4.5	3.4/5.2	7.8/3.6	11/4.8	2/2.4
3%	1/2.9	3/3.5	3/3.7	1/3.8	3/2.3	2/3.7	3/4.4	11.4/3.1	10.6/4.1	2/2.4
6%	1/2.6	2/3.3	3/2.9	1/3.1	3/2.2	2/3.3	2.4/3.2	8/2.5	8.4/3.6	2/2.3
cfgs	2.2/3.4	2.3/3.5	4.1/3.5	1.2/3.9	2.3/2.0	2.7/3.6	2.6/4.3	9.3/3.1	9.8/4.1	2/2.3

Thus, $\delta = 0$ denotes interpolation within demonstrations, while $\delta = 1$ places the query one demonstration-range width beyond the nearest boundary. We group queries into five regimes: interpolation ($\delta = 0$), near ($\delta \in (0, 0.5]$), moderate ($\delta \in (0.5, 2]$), far ($\delta \in (2, 5]$), and very-far extrapolation ($\delta \in (5, 8]$).

Functions used for evals: Experiments use ten data-generating functions G: *affine, quadratic, cubic, exponential, damped-sinusoidal, piecewise-linear, step, sawtooth, logistic* and *sinusoidal*. Their exact definitions and plots are provided in Appendix C. We added *sawtooth* and *sinusoid* to the testbench to show a different effect, where LLM mostly adapt a collapsed mode outside support mostly predicting along a constant line (collapsed mode of prediction).

4.1 Coexisting predictive modes

For each run, we sample a 100 demonstrations from each G and hold it fixed across queries to construct $F_{ICL}(\cdot|D)$. We also do ablations varying number of demonstrations with $n = 20, 40, 60, 80$ and 100. Prediction errors across, averaged over different sparsity runs, are reported in Appendix F. Increasing error can be mostly dependent on G with fast growing functions giving larger errors. This alone does not establish a change in predictive mode, since it can also reflect worsening extrapolation within a single predictive regime (single mode with growing noise null).

We use RANSAC peeling procedure with a deliberately simple hypothesis class consisting of a constant (line with slope zero), a line, and a low-degree polynomial (of degree two and three). We use a RANSAC tolerance of $\tau = 5\%$ of the local y-range, requiring at least 10 inliers to identify a mode. All experiments fix $n = 100$ and evaluate predictions across all δ regimes. We also fit with multiple inlier bands (more on RANSAC hyperparameter, Appendix D). Each run with a set of functions used to peel along with with fitting hyper-parameters form a configuration (*cfg*). We use thirty different *cfg*s (Details in Appendix D).

Figure 2 shows a smaller δ-range to emphasize geometric structure over $\hat{y}$. At this scale, the peeled modes account for essentially almost entire prediction surface: across the four panels, only 14 out of 427 pooled predictions ($\sim 3\%$) fall outside the tolerance band of every annotated mode. This near-complete coverage makes visually explicit that the predictions are not dispersed around a single function with noise, but concentrate around a small number of modes.

The same *cfg* is applied symmetrically to both the LLM prediction surface $F_{\mathrm{ICL}}(\cdot \mid D)$ and the ground-truth function G. The difference in fragmentation is the evidence to the branching of the prediction surface. Table 1 compares the number of peeled branches with the corresponding decomposition of G. The two significant fragmentation patterns can be seen contrasting a simple function like affine with periodic complex ones. Under most *cfg*s, the former is simply described by a line while the prediction surface fragments; while for oscillating Gs like *sawtooth* and *sinusoid* which allows $8 - 11$ fragments under most *cfg*s, the model can produce fewer branches by collapsing to simpler continuations.

We further do a local comparison to reveal the growing separation between LLM predictions and single-function controls as queries move beyond the demonstrations. Using a small striding window over the trace of $\hat{y}_j$ from 1 to m, within each small window of consecutive queries, we ask whether the predictions are better traced by one line with at most one break, or by two separate lines between which neighboring queries alternate. A window counts as branching only when the second description

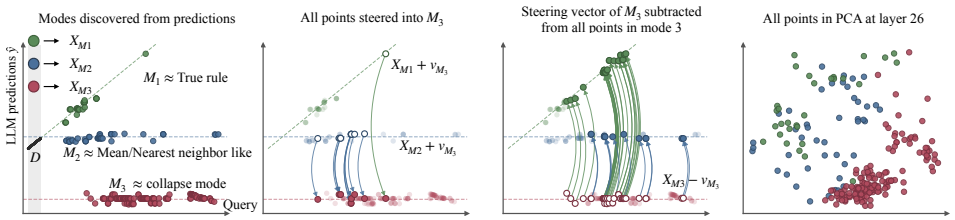

Figure 3: **Steering along a single residual-stream direction switches the prediction mode.** Qwen-2.5-7B, affine task, 100 demonstrations (D: black, in-support: shaded region). **(a)** Predictions on held-out queries fall onto three modes: M_1- near the true rule, M_2- a flat line ($\hat{y} \approx 30$), and M_3- a collapse to a constant ($\hat{y} \approx -277$). Dashed lines show best fit lines to each mode. **(b)** Adding the steering vector v_{M_3} at layer 26 ($\alpha = +0.5$) moves most queries from M_1 and M_2 into M_3. **(c)** Subtracting it ($\alpha = -1$) moves queries out of M_3, onto M_1 or M_2. Hollow markers are predictions before steering and filled markers after; arrows connect the same query. **(d)** First two principal components of the layer-26 residual stream at the last input token, coloured by baseline mode; queries in M_3 occupy a distinct region.

clearly wins, the two lines are well separated, and the predictions switch between them at least twice. Pooling across the tested demonstration counts, the fraction of windows exhibiting these branches rises from 4.9% in interpolation to 15.5%, 16.7%, 18.6%, and 19.4% across successive extrapolation regimes. In contrast, the corresponding rates remain below 0.4% for single-function controls with growing Gaussian or heavy-tailed noise, and below 1.2% for controls following a different smooth function. Thus, the tested forms of increasing noise and function mismatch do not reproduce the observed branch structure. Moreover, 53% of evaluated $F_{ICL}(\cdot|D)$ surfaces exhibit branches in at least two non-overlapping extrapolation windows.

We compare against four control surfaces, built on the same queries and demonstrations and scored by the same rule: (1) The true function G with no noise (2) G plus noise that grows with δ, scaled to match the LLM's own scatter (3) A single smooth curve fitted through the middle of the LLM's own predictions, plus small noise (4) G inside the demonstration range, bending to a flat plateau beyond it (approximate collapse/kernal behaviour of LLM). None of these has branches are flagged more than 3.6% of the time, while a fifth control surface with two planted branches is flagged 90% of the time. The LLM surfaces are flagged far above all four, so their branching is not explained by growing noise, by the LLM's overall drift from G, or by a single curve that flattens out. Appendix I reports the full analysis, control and failures.

Steering the discovered modes: Peeling provides a descriptive decomposition of the prediction surface, but does not establish that the recovered modes correspond to anything the model is internally doing. If the modes reflect genuine internal commitments on hypothesis rather than artifacts of how we chose to fit the surface, they should be amenable to causal intervention. Steering vector is computed using a subset of selected x_j^qs for each mode and evaluated on disjoint held-out set. We do two kinds of steering tests: (a) positive injection asks whether the vector acts as an attractor for a mode (whether a mode can be induced into a query that did not originally express it), and (b) negative injection asks what is left once a mode's direction is removed (weather a query when pushed out of the bin of a mode jumps intoa random position or another mode expressed at other queries).

For this analysis we use modes discovered by simple *cfg* of constants and lines with no polynomials on affine G for *Qwen-2.5-7B, Gemma 3-4B, Gemma 3-12B* and *Llama 3.1-8B*. As seen in Figure 3 adding steering vector derived from a mode can induce its associated behaviour, while subtracting a it can move outputs onto other discovered branches, including the mode corresponding to true G. In the four-model affine experiment, the percentage of points which got attracted to target mode range from 79-89$\%$ and removal rates from 63-77$\%$. The full results and experiment details are in Appendix E. These interventions show that the modes discovered is strongly related to the underlying hypothesis used by the LLM.

Conditional-Density Test for Multiple modes marginalized on demonstrations: We next ask whether evidence of multiple predictive modes also appears without fitting branches to the prediction surface. For each model, target function G, and query x^q, we sample 100 different demonstration sets, each containing $n = 100$ examples, and collect one greedy prediction per prompt. In this trial,

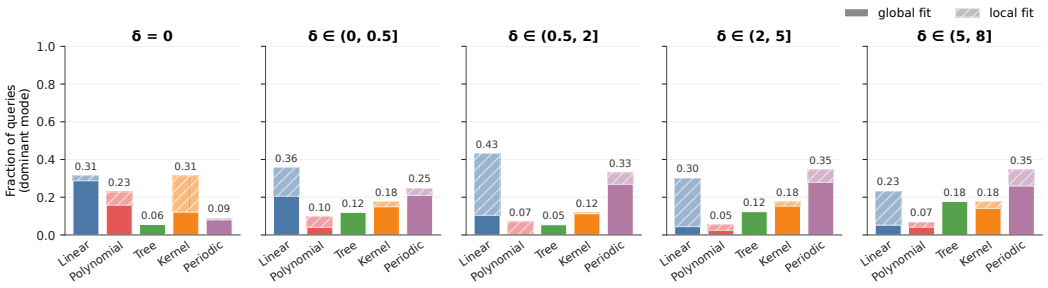

Figure 4: Attribution profiles across δ-bins for $G = \textit{sinusoid}$, aggregated over the eight LLMs and five demonstration densities. Each panel corresponds to one δ-bin; bars show the fraction of test queries whose dominant family falls within each of five hypothesis families. The active family set remains plural in every δ-bin. 'Linear' dominates near the demonstrated support and progressively localizes, whereas 'Periodic' emerges at far as the dominant globally fitted shape.

the model, target function, and query remain fixed, only the demonstrations change. We test whether these predictions form a single group or several distinct groups. Across three models, ten target functions, and fifty queries per target, the test rejects a single-group description at 53% of query locations, after correction for multiple testing and requiring a secondary group to contain at least 5% of predictions. This provides evidence of predictive multiplicity independent of the function class and fitting order used in peeling.

The result concerns predictions across different demonstration sets and by itself does not establish that changing the query changes their weights on hypothesis. Though the dependence on query distance also varies across tasks. We therefore use this test as complementary evidence for multiple prediction groups L.

4.2 Query-dependent predictive behavior

Pointwise attribution characterizes how these behaviors relate to classical inductive biases. We compare LLLM prediction with 34 classical algorithms fitted to the demonstrations. For analysis we group them as yielding five families: **Linear** (OLS, ridge, Lasso, elastic-net, Bayesian linear), **Polynomial** (polynomial regression, orthogonal-polynomial, splines), **Tree** (CART, random forest, gradient-boosted), **Kernel** (kernel ridge, SVR, GP, RBF network), and **Periodic** (Fourier features, periodic GP, locally-periodic GP). A *scope* axis distinguishes *global* fits (the algorithm fit on the full demonstrations from *local* fits (the same algorithm fit on the k nearest demonstrations to the query, `Local_*`). The two-axis decomposition lets the attribution profile express both *what* analytic shape the LLM expresses and *at what scope*. Aggregating across the eight models and five sparsities yields one profile per (G, δ) pair.

The analysis picks the best fit predictor as a simple *argmin* over the error of demonstration fitted predictors in approximating the LLM predictions over the queries. A tie aware analysis is also given in Appendix G along with full results. Reading the analysis left to right reveals the structured redistribution as predicted by the framework especially in far δ bins. For instance on the sinusoid G (Figure 4), As δ grows, two changes occur. First, the **Linear** family progressively localizes (its global-fit component shrinks) while its local-fit component grows. Second, the **Periodic** family, which carries only 0.09 of mass at $\delta = 0$, rises monotonically to 0.35 at $\delta \in (5, 8]$ and becomes the modal family at far and very-far extrapolation, expressed predominantly as a global fit. These attribution describes changes in predictive behavior but does not independently establish hypothesis switching.

Testing fixed probability weights. A change in the greedy answer need not imply a change in hypothesis weights: fixed weights can still yield different most likely answers across queries. We now directly test the proposed view on *Qwen2.5-7B-Instruct, Gemma-3-12B-it,* and *Llama-3.1-8B-Instruct* with all ten target Gs, using five demonstration sets of $n = 100$. For each D, 25 queries (we call them discovery queries) identify recurring patterns in the prediction, like constants and lines in the

numerical-answer distributions. Together with G and the mean demonstration output, these form a candidate family frozen before evaluation at 50 separate test queries. This discovery step supplies the hypotheses used specifically for the probability test.

For each test query, we measure how much probability the model assigns to answers near each candidate's prediction. We expand possible answer strings using the model's token probabilities and add together the probabilities of strings that represent the same numerical answer. Responses not reached by this computation remain unresolved, giving lower and upper bounds on the measured probabilities. We allow answers to vary around a candidate's prediction within an interval whose half-width is the larger of 5% of the demonstration output range and 5% of the prediction magnitude. To test the fixed-mixture account, we assume that each candidate places at least 95% of its numerical-answer probability inside its interval, while allowing additional error for answers the candidate family does not explain. These assumptions translate the measured answer probabilities into bounds on each candidate's weight. If the minimum weight required at one query exceeds the maximum allowed at another, the candidate cannot have the same weight at both queries. We make this comparison only where candidate intervals do not overlap and exclude queries with more than 0.20 probability mass on numerical answers outside the family. The complete probability accounting and bounds are given in Appendix J.

With demonstrations held fixed, a single set of hypothesis weights cannot explain $2{,}090$ of $3{,}340$ tested pairs of extrapolation queries with different greedy predictions in Qwen (63%), and 702 of $2{,}019$ pairs in Gemma (35%). Thus, for these comparisons, the changing greedy answer is accompanied by a change in probability support that exceeds the allowed uncertainty and error. At $n = 100$, three of the thirty model-G combinations are classified as showing a change in which hypothesis has the greatest probability weight.

Though this shows strong evidence for the proposed hypothesis, this results should be read along with the robustness check. The number of positive classifications also depends substantially on the prediction intervals. For demonstration counts ($n = 20, 100$), widening the interval scale from 2% to 5% to 10% increases the number classified as showing reweighting and dominant switching from $2/60$ to $6/60$ to $17/60$. For the tested *Qwen-affine* case, changing the discovery queries also changes the discovered hypotheses. The conclusions therefore depend on both the candidate family and the tolerance used to associate answers with its predictions.

Scope under known training distributions: Finally, we ask whether moving beyond the demonstrations is enough to produce different predictive modes, or whether this also depends on what the model learned during training. Following Garg et al. (2022), we train small GPT-2 like transformers on either linear functions alone or a mixture of linear and sinusoidal functions. We then provide demonstrations from a small part of the training range and query farther away. Models trained only on lines generally continue the linear relation. When models trained on both function classes receive demonstrations that could fit either, their predictions mostly fall between the two possible continuations. Repeatedly drawing new demonstrations rarely produces two clearly separated groups of predictions. The larger changes in behavior occur when queries leave the training range, rather than simply the demonstration range. The takeaway is that multi-mode when expressed is seen inside the training range. Full results are in Appendix K.

5 Conclusion

We reframe in-context learning as case-based, transductive adaptation: demonstrations induce multiple predictive modes, and the query determines which is expressed. Across eight LLMs and ten regression tasks, peeling, steering, pointwise attribution converge on this picture: prediction surfaces decompose into coexisting modes, those modes correspond to distinct directions in the geometry of representation in the residual-stream, and the dominant inductive bias redistributes systematically with query–demonstration similarity. This lens reconciles meta-learning, Bayesian, and kernel accounts of ICL by relocating their disagreements from the prompt to the query, and recasts ICL as a dense, controlled instantiation of the query-conditioned hypothesis selection long observed in human function learning. Practically, it implies that a small amount of query-level supervision is sufficient to identify and steer desired modes at inference time without prompt engineering or parameter updates, turning the apparent jaggedness of ICL into a controllable behavior.

AI USE STATEMENT

This work is a study of LLM and hence most experiments involve running LLMs. Beyond that, this work used generative AI tools for assisting in coding and polish the writing. Experiments were designed without using AI. Coding tools were used for assistance like filling in function body which were verified by one of the authors. Specifically, AI tool were used extensively to manage SLURM jobs and schedules the experiments to help expedite execution. The shell scripts written for this were also manually verified by one of the authors. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, or interpret results, formulate mathematical claims. Additionally, we used generative AI tools to better some of the sentences for readability which was verified by author one sentence at a time. We also used AI tool as search engines for finding related work to read. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or crowdsourced annotation. All demonstrations and queries are drawn from synthetic, data-generating functions, and all evaluated models are publicly released open-weight LLMs used under their respective licenses, with no fine-tuning or parameter updates. The small transformers in are trained from scratch on synthetic data. We do not foresee direct harmful applications: the activation-steering results concern numerical regression prompts and are intended as evidence about the internal structure of in-context learning, not as a method for altering model behavior on deployed tasks. The main cost of the study is compute (Appendix I); we report the full budget in so that the environmental footprint can be assessed. The authors declare no conflicts of interest or sponsorship that could bias the results.

REPRODUCIBILITY STATEMENT

All experiments use publicly available open-weight models (Section 4) and synthetic data-generating functions whose exact definitions are given in Table 3 (Appendix C). The prompt template, chat-template wrapping, greedy decoding settings, per-model token budgets, and the trace-stripping and number-extraction rules are specified in Appendix B, together with parse success rates (Table 2). The RANSAC peeling procedure is given as pseudocode (Algorithms 1 and 2), with all fixed hyperparameters in Table 4, the full 30-configuration sweep in Table 5, and per-cell seeding derived deterministically from a global seed (Appendix D). Hardware and compute budgets are reported in Appendix I. Source code for the inference sweep, peeling, attribution, steering, and all statistical tests, together with the cached prediction pools, will be made public post acceptance.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37:129696–129742, 2024.

Anas Awadalla, Mitchell Wortsman, Gabriel Ilharco, Sewon Min, Ian Magnusson, Hannaneh Hajishirzi, and Ludwig Schmidt. Exploring the landscape of distributional robustness for question answering models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5971–5987, 2022.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jerome R. Busemeyer, Eunhee Byun, Edward L. DeLosh, and Mark A. McDaniel. Learning functional relations based on experience with input–output pairs by humans and artificial neural networks. In Koen Lamberts and David R. Shanks (eds.), *Knowledge, Concepts, and Categories*, Studies in Cognition, pp. 405–437. Psychology Press, Hove, UK, 1997.

J. Douglas Carroll. Functional learning: The learning of continuous functional mappings relating stimulus and response continua. Research Bulletin RB-63-26, Educational Testing Service, Princeton, NJ, 1963.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Edward L. DeLosh, Jerome R. Busemeyer, and Mark A. McDaniel. Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(4):968–986, 1997. doi: 10.1037/0278-7393.23.4.968.

Gilad Deutch, Nadav Magar, Tomer Natan, and Guy Dar. In-context learning and gradient descent revisited. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1017–1028, 2024.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Michael A. Erickson and John K. Kruschke. Rules and exemplars in category learning. *Journal of Experimental Psychology: General*, 127(2):107–140, 1998. doi: 10.1037/0096-3445.127.2.107.

Fabian Falck, Ziyu Wang, and Chris Holmes. Is in-context learning in large language models bayesian? a martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 12784–12805, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.

Eghbal A Hosseini, Yuxuan Li, Yasaman Bahri, Declan Campbell, and Andrew Kyle Lampinen. Context structure reshapes the representational geometry of language models. *arXiv preprint arXiv:2601.22364*, 2026.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Michael L. Kalish, Stephan Lewandowsky, and John K. Kruschke. Population of linear experts: Knowledge partitioning and function learning. *Psychological Review*, 111(4):1072–1099, 2004a. doi: 10.1037/0033-295X.111.4.1072.

Michael L Kalish, Stephan Lewandowsky, and John K Kruschke. Population of linear experts: knowledge partitioning and function learning. *Psychological review*, 111(4):1072, 2004b.

Kyunghee Koh and David E. Meyer. Function learning: Induction of continuous stimulus-response relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17(5):811–836, 1991. doi: 10.1037/0278-7393.17.5.811.

Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. *arXiv preprint arXiv:2505.00661*, 2025.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021.

Stephan Lewandowsky, Michael Kalish, and Sai K. Ngang. Simplified learning in complex situations: Knowledge partitioning in function learning. *Journal of Experimental Psychology: General*, 131 (2):163–193, 2002. doi: 10.1037/0096-3445.131.2.163.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022.

Christopher G. Lucas, Thomas L. Griffiths, Joseph J. Williams, and Michael L. Kalish. A rational model of function learning. *Psychonomic Bulletin & Review*, 22(5):1193–1215, 2015. doi: 10.3758/s13423-015-0808-5.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. In-context learning with retrieved demonstrations for language models: A survey. *Transactions on Machine Learning Research*, 2024.

Mark A. McDaniel and Jerome R. Busemeyer. The conceptual basis of function learning and extrapolation: Comparison of rule-based and associative-based models. *Psychonomic Bulletin & Review*, 12(1):24–42, 2005. doi: 10.3758/BF03196347.

Mark A. McDaniel, Michael J. Cahill, Mathew Robbins, and Chelsea Wiener. Individual differences in learning and transfer: Stable tendencies for learning exemplars versus abstracting rules. *Journal of Experimental Psychology: General*, 143(2):668–693, 2014. doi: 10.1037/a0032963.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Competition dynamics shape algorithmic phases of in-context learning. In *International Conference on Learning Representations*, 2025.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in neural information processing systems*, 36:14228–14246, 2023.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, 2022.

Eric Schulz, Joshua B. Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel J. Gershman. Compositional inductive biases in function learning. *Cognitive Psychology*, 99:44–79, 2017. doi: 10.1016/j.cogpsych.2017.11.002.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent? In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Robert Vacareanu, Vlad Andrei Negru, Vasile Suciu, and Mihai Surdeanu. From words to numbers: Your large language model is secretly a capable regressor when given in-context examples. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=LzpaUxcNFK.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Daniel Wurgaft, Ekdeep Singh Lubana, Core Francisco Park, Hidenori Tanaka, Gautam Reddy, and Noah D. Goodman. In-context learning strategies emerge rationally. *arXiv preprint arXiv:2506.17859*, 2025.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *ICLR*, 2022.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Yuxiang Zhou, Jiazheng Li, Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14365–14378, 2024.

A TRANSDUCTIVE LEARNING IN HUMANS

Motivation from Human Function Learning: The framing developed in this work, that in-context predictions arise from query-conditioned selection among multiple coexisting hypotheses rather than from a single fixed learner, is not new to the study of inductive inference. Half a century of research on how humans generalize from limited input–output demonstrations has converged on essentially the same picture. Early theories cast learners as either estimating explicit parametric forms Carroll (1963); Koh & Meyer (1991) or as performing similarity-based association over training exemplars (Busemeyer et al., 1997), and neither account on its own proved sufficient: humans extrapolate in ways that pure exemplar models cannot produce (DeLosh et al., 1997), yet show locality and context-sensitivity that pure rule models cannot explain (Erickson & Kruschke, 1998). The resolution, established across behavioral, computational, and rational analyses, is that learners maintain multiple candidate hypotheses simultaneously, and that the hypothesis expressed at a given query is determined jointly by the demonstrations and the query itself Kalish et al. (2004a); McDaniel & Busemeyer (2005); Lucas et al. (2015); Schulz et al. (2017). This recurring finding, robust across rule-based, associative, and Bayesian formulations and across interpolation, extrapolation, and compositional generalization, suggests that query-conditioned hypothesis selection is a general property of inductive inference under limited data rather than an artifact of any particular learner. We treat this as the conceptual motivation for analyzing in-context learning in the same terms.

Cognitive accounts of query-conditioned selection: Three converging accounts in cognitive science explain why query-conditioned selection arises in human learners, each emphasizing a different mechanism but agreeing on the basic structure. *Knowledge partitioning* accounts hold that learners represent complex tasks as collections of locally valid sub-models rather than a single integrated representation, with the active sub-model determined by stimulus context (Lewandowsky et al., 2002; Kalish et al., 2004a). *Hybrid rule-and-similarity* accounts hold that learners maintain rule-based and exemplar-based representations in parallel, with arbitration between them governed by the query and by individual disposition (Erickson & Kruschke, 1998; McDaniel & Busemeyer, 2005; McDaniel et al., 2014). *Compositional* accounts hold that the hypothesis space itself is generated by combining simpler primitives, so that different compositions dominate in different functional regions (Schulz et al., 2017). The mechanisms differ; the prediction is the same. A single global hypothesis is insufficient, and the hypothesis expressed at a query reflects an interaction between the demonstrations and the query rather than a fixed mapping induced by the demonstrations alone.

Why query-conditioned selection is rational. Query-conditioned selection is not merely descriptive of human behavior; it is the expected behavior of a Bayesian learner whose prior places mass on multiple plausible hypothesis classes. Under such a prior, the posterior predictive at a query is a weighted average of predictions from each hypothesis in which the weights depend on query through the predictive density of each hypothesis at that query (Lucas et al., 2015; Schulz et al., 2017). When the demonstrations leave several hypotheses approximately consistent, the weights are diffuse globally but vary locally: hypotheses that assign high density to a region of input space dominate predictions in that region, and the dominant hypothesis shifts as the query moves through the input space. Committing to a single hypothesis is then strictly suboptimal whenever the posterior is not concentrated, and query-conditioned mixing is the rational response to genuine hypothesis ambiguity under limited data (Lucas et al., 2015). This rational analysis links the cognitive findings to the broader principle that learners with structured priors should express different inductive biases at different queries.

Behavioral evidence for query-conditioned hypothesis expression: The evidence for this picture comes from a sequence of behavioral studies that progressively sharpened the empirical signature. DeLosh et al. (1997) trained participants on linear, exponential, and quadratic input-output relationships and showed that participants extrapolated approximately linearly even after learning nonlinear functions, establishing that the hypothesis expressed in extrapolation regions differed systematically from the one expressed in interpolation regions; this observation motivated the model in which similarity-based prediction governs interpolation and a linear rule governs extrapolation (McDaniel & Busemeyer, 2005). Kalish et al. (2004a) strengthened the claim by collecting repeated responses to identical test stimuli and showing that response distributions in transfer regions were *multimodal*, with peaks corresponding to distinct underlying linear hypotheses.

This is a much sharper signature than aggregate response curves: a single learner with stochastic noise produces unimodal distributions at every query, and bimodality is a direct fingerprint of latent hypothesis discreteness. Lucas et al. (2015) re-examined a wide range of human function learning data with a mixture-of-kernels prior and showed that human predictions across interpolation, extrapolation, and transfer are well described by multiple hypothesis classes whose dominant member shifts as a function of the query. Schulz et al. (2017) extended this to compositional hypothesis spaces, showing that humans extrapolate sophisticated functions including compositions of linear, periodic, and smooth components, and that their predictions are best explained by a structured grammar over kernel combinations rather than any single kernel; the grammar selected varies across functional regions, again supporting query-conditioned hypothesis expression. Across these studies the methodology is consistent: characterize behavior at multiple queries under fixed demonstrations, look for systematic shifts in the expressed hypothesis as the query varies, and interpret those shifts in terms of an explicit vocabulary of candidate hypotheses.

Scaling up the analysis to in-context learning. The human function learning literature established query-conditioned hypothesis selection as a behavioral phenomenon, but its empirical resolution has been bounded by what behavioral data permit. Human experiments yield few responses per participant per query, force aggregation across participants to obtain stable estimates, and admit only a small number of test queries per session (DeLosh et al., 1997; Kalish et al., 2004a; Lucas et al., 2015). These constraints have meant that query-conditioned hypothesis expression has been inferred indirectly: from aggregate response curves, from distributional signatures collected over modest numbers of repetitions, or from posterior fits at the participant or group level rather than at individual queries (Kalish et al., 2004a; Lucas et al., 2015; Schulz et al., 2017). LLMs remove these constraints. The same demonstration set can be paired with arbitrarily many queries, the input space can be densely covered including far-extrapolation regimes that are difficult to probe behaviorally, and the prediction surface $\mathcal{F}_{\text{ICL}}(\cdot \mid \mathcal{D}^{\text{train}})$ can be characterized at a resolution that human studies cannot match. Our methodological choices, peeling against a deliberately narrow hypothesis class to detect mode multiplicity, pointwise attribution against a library of classical inductive biases to identify the expressed hypothesis, and stratification by query-demonstration similarity to expose query-conditioned switching, are direct descendants of the human function learning tradition, lifted to a setting in which the mapping from demonstrations to predictions is dense, controlled, and reproducible. In this sense the analysis we develop is not a new explanatory framework so much as a high-resolution instantiation of an existing one, applied to a learner whose behavior is amenable to the kind of dense, query-resolved probing the behavioral literature has long sought.

B Prompt template, decoding, and answer extraction

Prompt template. Every prompt presented to an LLM consists of the demonstration set $\mathcal{D}^{\text{train}}$, a single test query x^{test}, and a fixed instruction. The user-facing message is

```
Predict the y value of the target x.

Here are observed (x, y) pairs:

x_1, y_1;
x_2, y_2;
...
x_n, y_n;

The x-value to predict y for is:
x_test, ?;

Print only the best estimate number (float) as the answer
and nothing else. Do not print extra words or full sentences,
only the number.
```

Each (x_i, y_i) pair is formatted to two decimal places. Demonstrations are presented in a randomised order under a per-cell deterministic seed, so that position within the prompt is uncorrelated with prox-

imity of any demonstrated input to the test query. The user message is wrapped in each model's chat template via `tokenizer.apply_chat_template` with `add_generation_prompt=True`, and the model's completion under greedy decoding (`do_sample=False`) is recorded as y^{LLM}.

Decoding budget. The maximum number of generated tokens is set per model family: 1024 for non-reasoning models (Llama-3.x, Qwen2.5, Gemma-3) and 16384 for the two reasoning models in our test-bench (DeepSeek-R1-Distill-Qwen-14B and GPT-OSS), which interleave a chain-of-thought trace before the final answer.

Why naive parsing fails on reasoning models. For non-reasoning models the completion is a single number (possibly with surrounding whitespace or quotes), and a direct `float` cast succeeds. For reasoning models the completion contains a sequential trace followed by the final answer, and parsing the first numeric token would consistently extract a number used in the reasoning rather than the model's conclusion. The two reasoning models in our test-bench use different trace formats and each requires its own boundary marker.

DeepSeek-R1. DeepSeek-R1-Distill emits an explicit reasoning block delimited by `<think>...</think>`, followed by the final answer. A typical completion is

```
<think>
Okay, so I need to predict y for x = 51.96. Looking at the
demonstrations, the relationship is approximately linear with
slope around 2 ... let me check: for x=-1.82, y=-7.56, ratio
~4.15 ... [~1500 tokens of reasoning] ...
</think>

103.92
```

The `</think>` marker reliably separates trace from answer.

GPT-OSS. GPT-OSS uses *channel* decoding: the model emits a structured sequence of `<|channel|>analysis<|message|>...` followed by `<|channel|>final<|message|>...`, where only the `final` channel is intended for the user. When the completion is detokenised with special tokens removed (the default in our pipeline), the boundary appears as the contiguous string `assistantfinal`; when special tokens are preserved, the boundary appears as `<|channel|>final`. We match either form. A typical completion is

```
analysisWe have a dataset of (x, y) pairs. We need to predict y
for x=51.96. The data seems to be linear? Let's examine pattern:
For x=-1.82, y=-7.56 ratio ~4.15. For x=3.23, y=5.08 ...
[~1500 tokens of analysis-channel reasoning]
assistantfinal103.92
```

Splitting on the rightmost `assistantfinal` (or `<|channel|>final`) leaves only the answer span.

Trace-stripping rule. We strip the trace prefix by searching the case-folded completion for the rightmost occurrence of any of the following markers and discarding everything up to and including that marker:

```
"assistantfinal"      # GPT-OSS, special tokens stripped
"<|channel|>final"    # GPT-OSS, special tokens preserved
"</think>"            # DeepSeek-R1
"final answer:"       # generic
"final answer"        # generic
"answer:"             # generic
"the answer is"       # generic
```

Table 2: Output-parse success rate (%) per ground-truth function G and model, pooled over the five δ-bins and five demonstration densities (1 250 queries per entry). For the non-reasoning models a query counts as parsed when the completion yields a number. For DEEPSEEK-R1-DISTILL-14B (greedy decoding, 16 384-token budget) a query counts as parsed only when the reasoning trace reaches `</think>` and the final answer parses; traces that hit the budget contribute no prediction and are excluded from all downstream analyses. Its rate decreases with the number of demonstrations, from 91% at $n_{\text{demos}} = 20$ to $52-54\%$ at $n_{\text{demos}} \in \{80, 100\}$, because longer contexts induce longer chains of thought that either loop or fail to commit to an answer within the budget.

$\mathcal{G}$	Llama3.2-3B	L3.1-8B	Qwen2.5-7B	Q2.5-14B	Gemma3-4B	G3-12B	DSeek-14B	GPT	mean
Affine	100.0	100.0	100.0	100.0	100.0	100.0	74.3	100.0	96.8
Quadratic	100.0	100.0	100.0	100.0	100.0	100.0	59.6	100.0	95.0
Cubic	100.0	100.0	100.0	100.0	100.0	100.0	59.4	100.0	94.9
Sinusoid	100.0	100.0	100.0	100.0	100.0	100.0	68.9	100.0	96.1
Logistic	100.0	100.0	100.0	100.0	100.0	100.0	71.9	100.0	96.5
P-linear	100.0	99.7	100.0	98.8	100.0	100.0	59.3	100.0	94.7
Exp.	100.0	100.0	100.0	100.0	100.0	100.0	58.9	100.0	94.9
Step	100.0	99.4	100.0	100.0	100.0	100.0	97.1	100.0	99.6
Sawtooth	100.0	100.0	100.0	100.0	100.0	100.0	72.4	100.0	96.5
D-sine	100.0	100.0	100.0	100.0	100.0	100.0	64.4	100.0	95.5
mean	100.0	99.9	100.0	99.9	100.0	100.0	68.6	100.0	96.1

If no marker is present the completion is left unchanged. Selecting the *rightmost* occurrence is critical: a verbose reasoning trace may quote any of these phrases mid-deliberation, but the actual final answer always follows the last occurrence.

Number extraction. After stripping, a three-stage regex extractor recovers the numeric prediction. (i) A line-anchored float at the start of any line of the stripped tail captures the clean-answer case (the model emits the number on its own line). (ii) An any-position scan over the stripped tail returns the *last* number, capturing the model's conclusion when it is not on a clean line (e.g. `"so y ≈ 103.92"`). (iii) An any-position scan over the *full* pre-strip text provides a final fallback for completions where the marker-based split removed too much. If none of the three returns a finite float, the cell records the prediction as missing. Across the full sweep the third fallback is invoked on $<1\%$ of completions, almost all of them edge cases in which the model produced no numeric output at all (e.g. a verbal explanation that the question is ill-posed). The full parse success is in Table 2.

C DATA-GENERATING PROCESSES AND PER-(MODEL, G) PREDICTIONS

C.1 FUNCTION DEFINITIONS

The ten data-generating processes (DGP, referenced as G in main text) used throughout the experiments are chosen to span a range of analytic shape classes: linear, polynomial, monotonic-saturating, oscillatory, and discontinuous. Coefficients are tuned so that $\max|y| \approx 25$ on the demonstration support $x \in [-10, 10]$, ensuring comparable scales across G in the in-support regime. Demonstrations and in-support test queries are drawn from this support; extrapolation queries are drawn beyond it according to the δ-binning of §3.3. Demonstrations are noise-free in all main-text experiments.

C.2 GROUND-TRUTH SHAPES

Figure 5 plots $G(x)$ over the demonstration support for each of the ten G. The shapes span smoothly varying functions (AFFINE, QUADRATIC, CUBIC), oscillatory functions (SINUSOID, DAMPED_SINE), saturating functions (LOGISTIC, EXPONENTIAL), piecewise-smooth functions (PIECEWISE_LINEAR, SAWTOOTH), and a discontinuous step.

C.3 PER-(MODEL, G) PREDICTIONS ACROSS DEMONSTRATION DENSITIES

Figures 6–10 present the raw LLM predictions y^{LLM} as a function of x^{test} for every (model, G) pair at each of the five demonstration densities $n_{\text{demos}} \in \{20, 40, 60, 80, 100\}$. In every figure, rows correspond to data-generating processes and columns to LLMs; grey markers show the n_{demos} demonstrations and colored dots show the LLM's prediction at each test query, colored by its δ-bin

G	Definition
AFFINE	$y = 2.5\,x - 3$
QUADRATIC	$y = 0.25\,x^2 - 5$
CUBIC	$y = 0.025\,x^3$
SINUSOID	$y = 25\sin(x)$
LOGISTIC	$y = \dfrac{25}{1 + e^{-x}} - 12.5$
PIECEWISE_LINEAR	$y = 25 - 2.5\,\lvert x\rvert$
EXPONENTIAL	$y = 25\,\dfrac{e^{0.3\,x} - 1}{e^3 - 1}$
STEP	$y = \begin{cases} -12.5 & x < 0 \\ +12.5 & x \geq 0 \end{cases}$
SAWTOOTH	$y = 25\,\mathrm{frac}(x/5) - 12.5$
DAMPED_SINE	$y = 25\,e^{-0.1\,x}\sin(x)/Z, \quad Z = \max_{x\in[-10,10]} \lvert e^{-0.1\,x}\sin(x)\rvert$

Table 3: Data-generating processes. Domain $x \in [-10, 10]$ for demonstrations and in-support test queries.

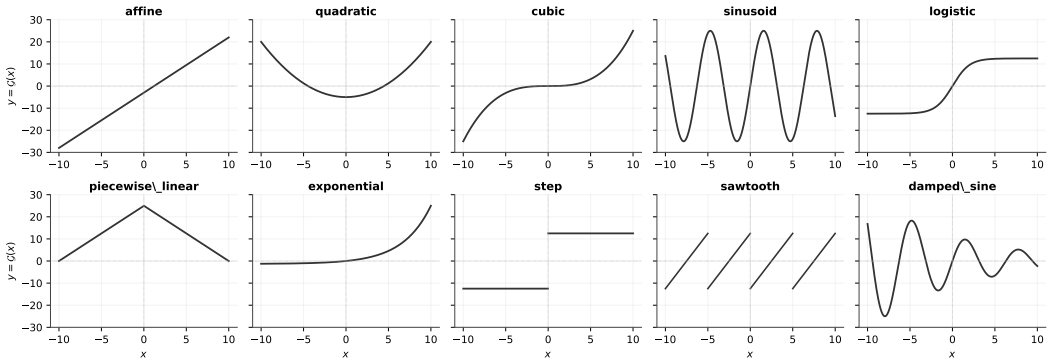

Figure 5: Ground-truth function $y = G(x)$ for each of the ten data-generating processes over the demonstration support $x \in [-10, 10]$.

(darkest = interpolation, brightest = very-far extrapolation). The ground-truth function $G(x)$ for each row is plotted separately in Figure 5; we omit it from the prediction panels so that the structure of the LLM output is not obscured. Per-panel y-axes are autoscaled so that small-amplitude modes (e.g., the boundary plateau on STEP and LOGISTIC) are not visually flattened by large-amplitude outliers in the same row.

The figures make the claims of §4 directly visible. (i) For in-support queries (darkest dots), predictions lie on or near $G(x)$ for the larger non-reasoning models on every G - demonstrations *can* pin down the task locally. (ii) As δ grows (lighter colours), predictions typically split into discrete tracks, each corresponding to a different surrogate mode, visually realising the multi-modal decomposition that peeling and point-wise attribution detect quantitatively. (iii) The set of surrogate tracks active in the far-δ region differs systematically with G: a linear extrapolation track dominates on CUBIC and EXPONENTIAL, periodic continuation dominates on SINUSOID and DAMPED_SINE, and a near-constant plateau at boundary demonstration values dominates on STEP and LOGISTIC. (iv) Comparing across figures shows that demonstration density modulates the *sharpness* of these surrogate tracks but rarely changes *which* ones are active/consistent with the Section 4 finding that the active subset of families is governed by G and δ rather than by n_{demos}.

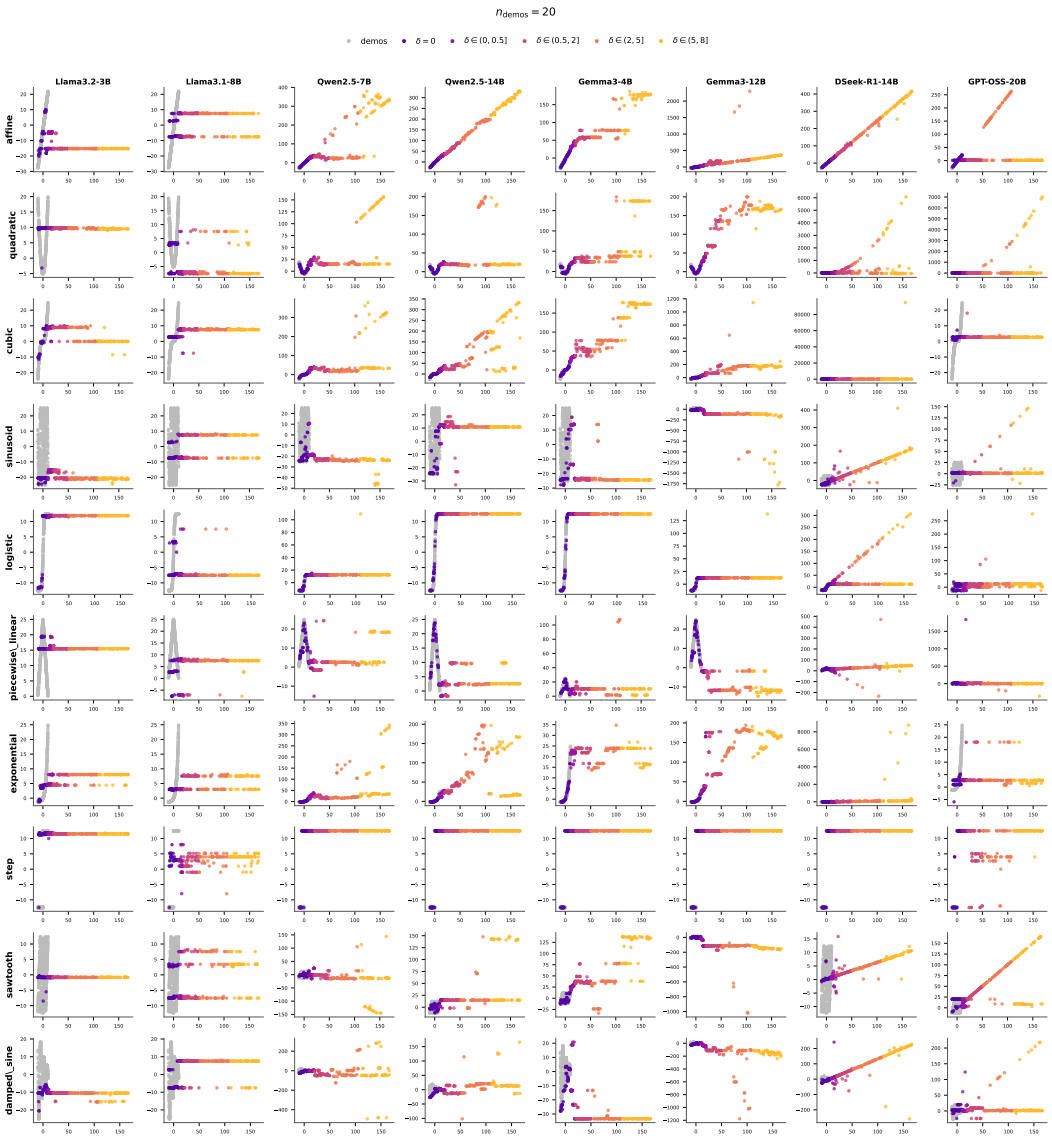

Figure 6: LLM predictions across (model, G) at $n_{\text{demos}} = 20$. Rows: AFFINE, QUADRATIC, CUBIC, SINUSOID, LOGISTIC, PIECEWISE_LINEAR, EXPONENTIAL, STEP, SAWTOOTH, DAMPED_SINE. Columns: LLAMA3.2-3B, LLAMA3.1-8B, QWEN2.5-7B, QWEN2.5-14B, GEMMA3-4B, GEMMA3-12B, DSEEK-R1-14B, GPT-OSS-20B. Grey: demonstrations. Coloured dots: LLM predictions, coloured by δ-bin (dark $= \delta = 0$, bright $= \delta \in (5, 8]$).

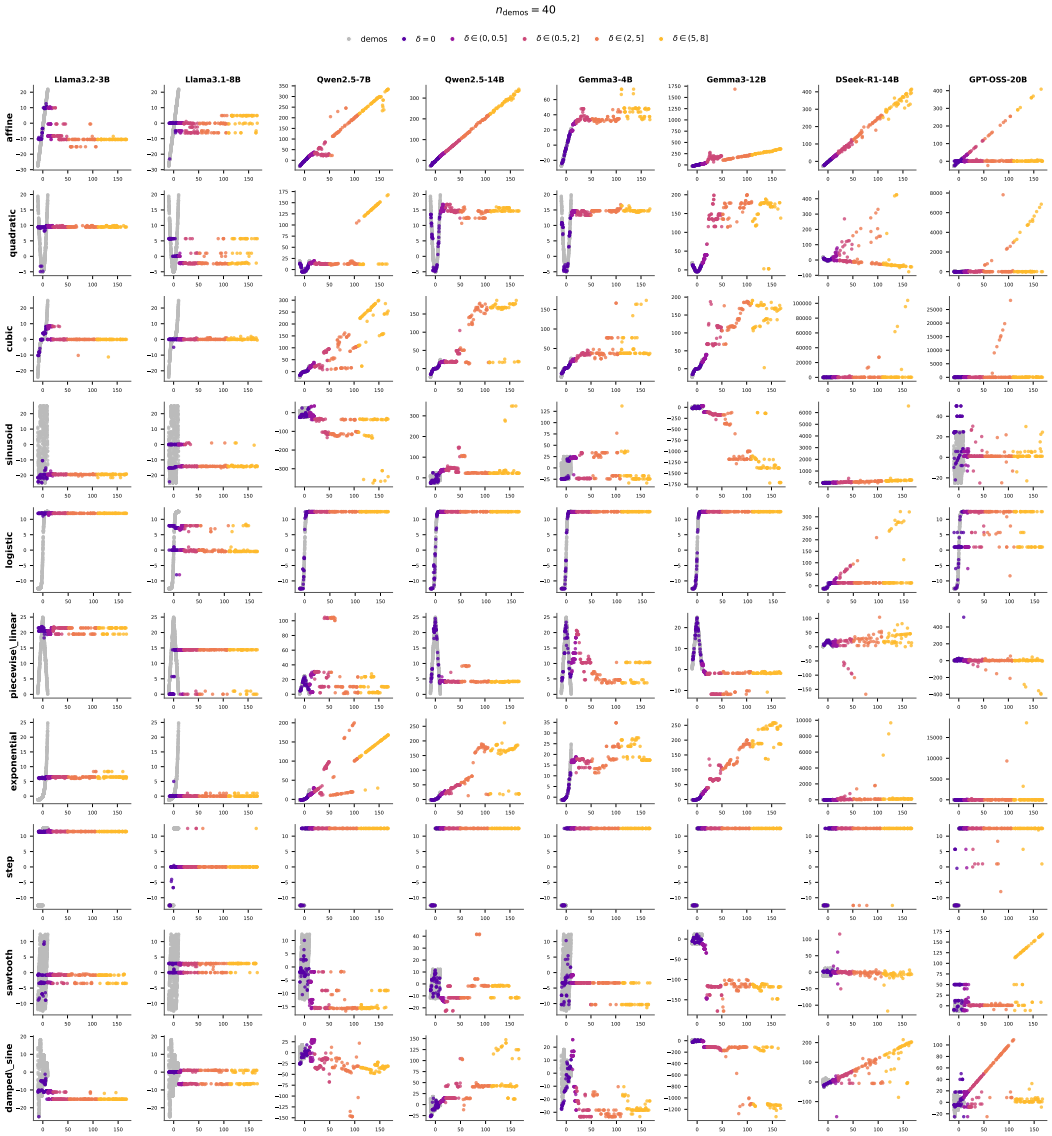

Figure 7: LLM predictions across (model, G) at $n_{\text{demos}} = 40$. Conventions as in Figure 6.

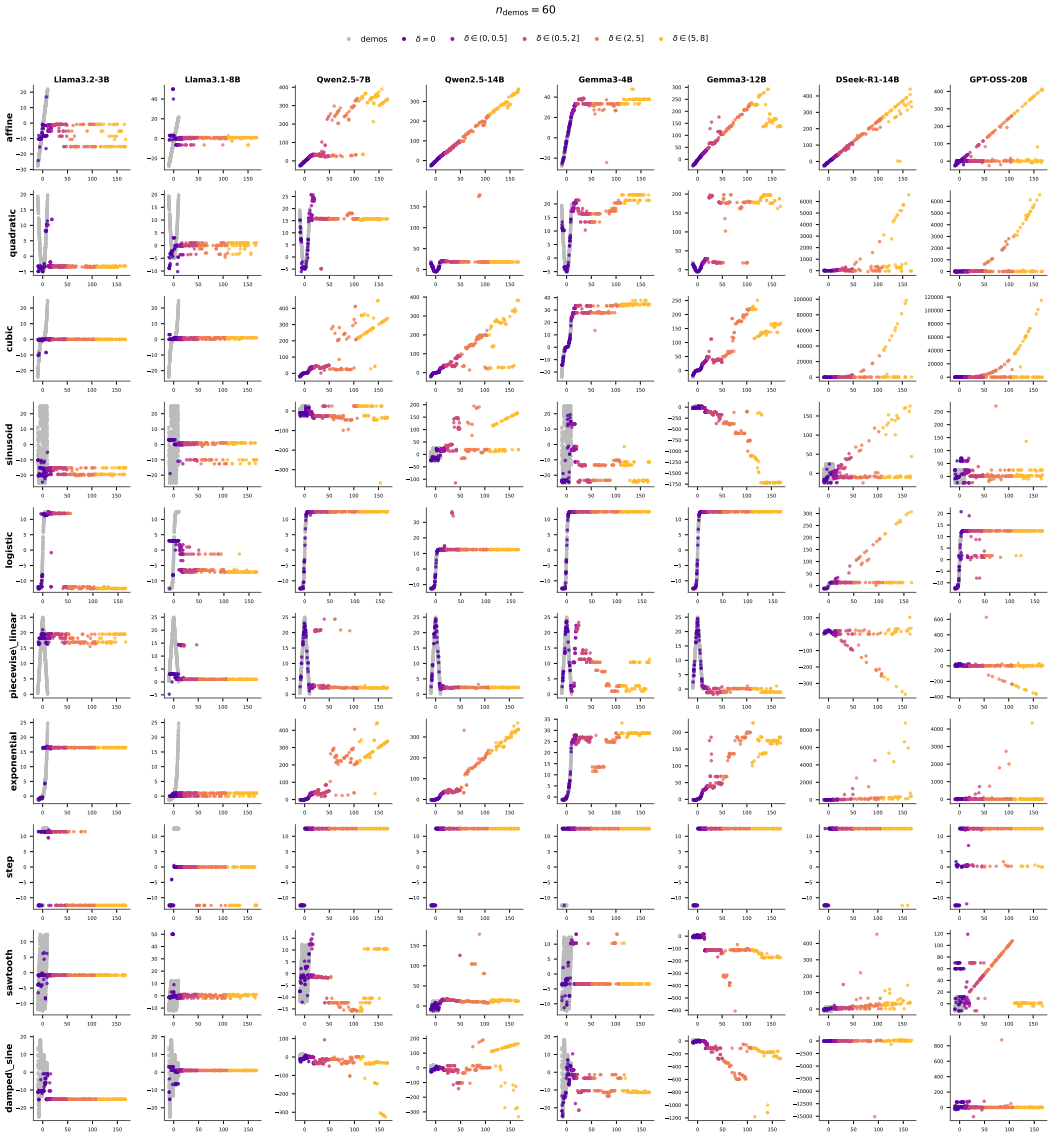

Figure 8: LLM predictions across (model, G) at $n_{\text{demos}} = 60$. Conventions as in Figure 6.

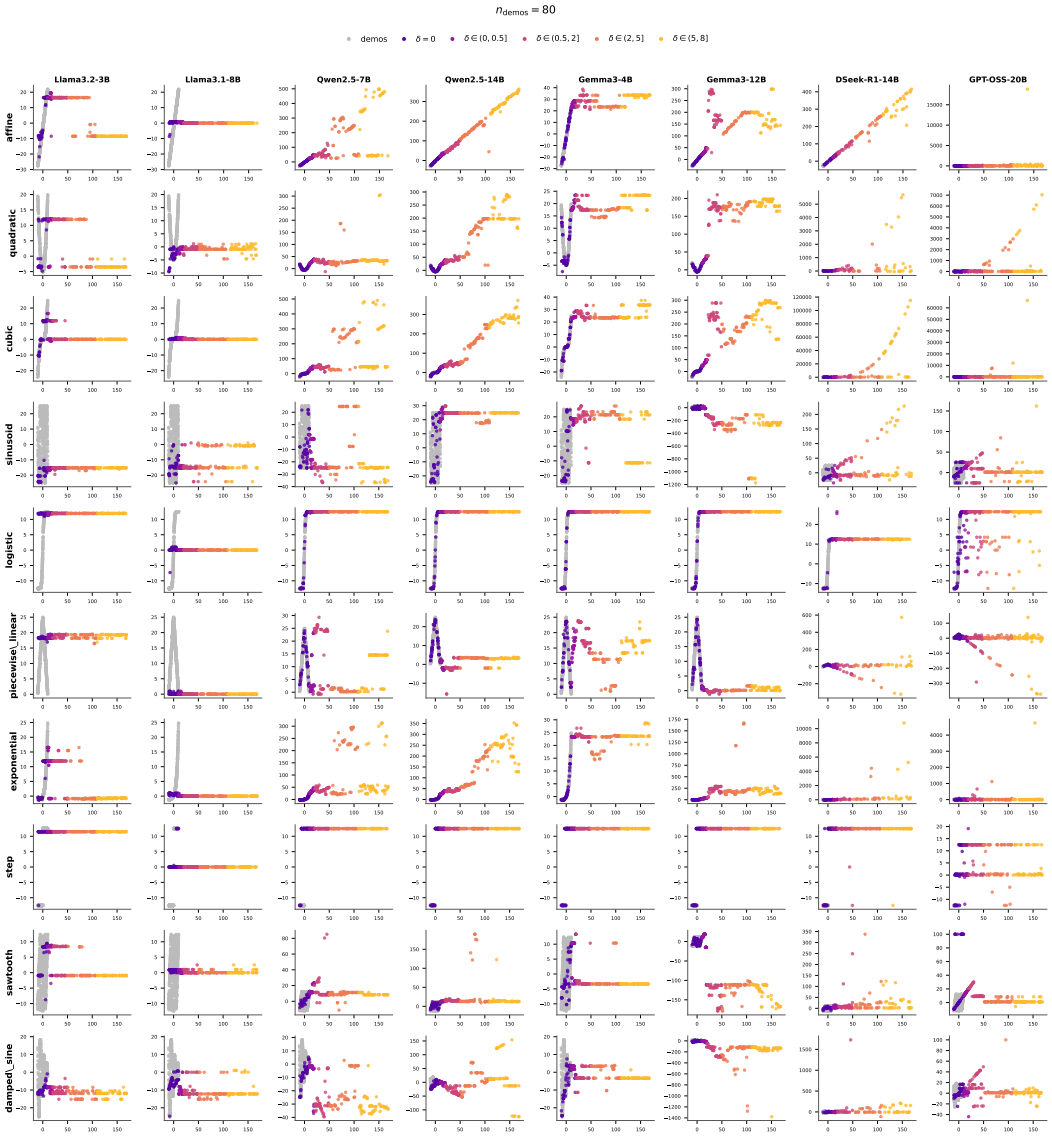

Figure 9: LLM predictions across (model, G) at $n_{\text{demos}} = 80$. Conventions as in Figure 6.

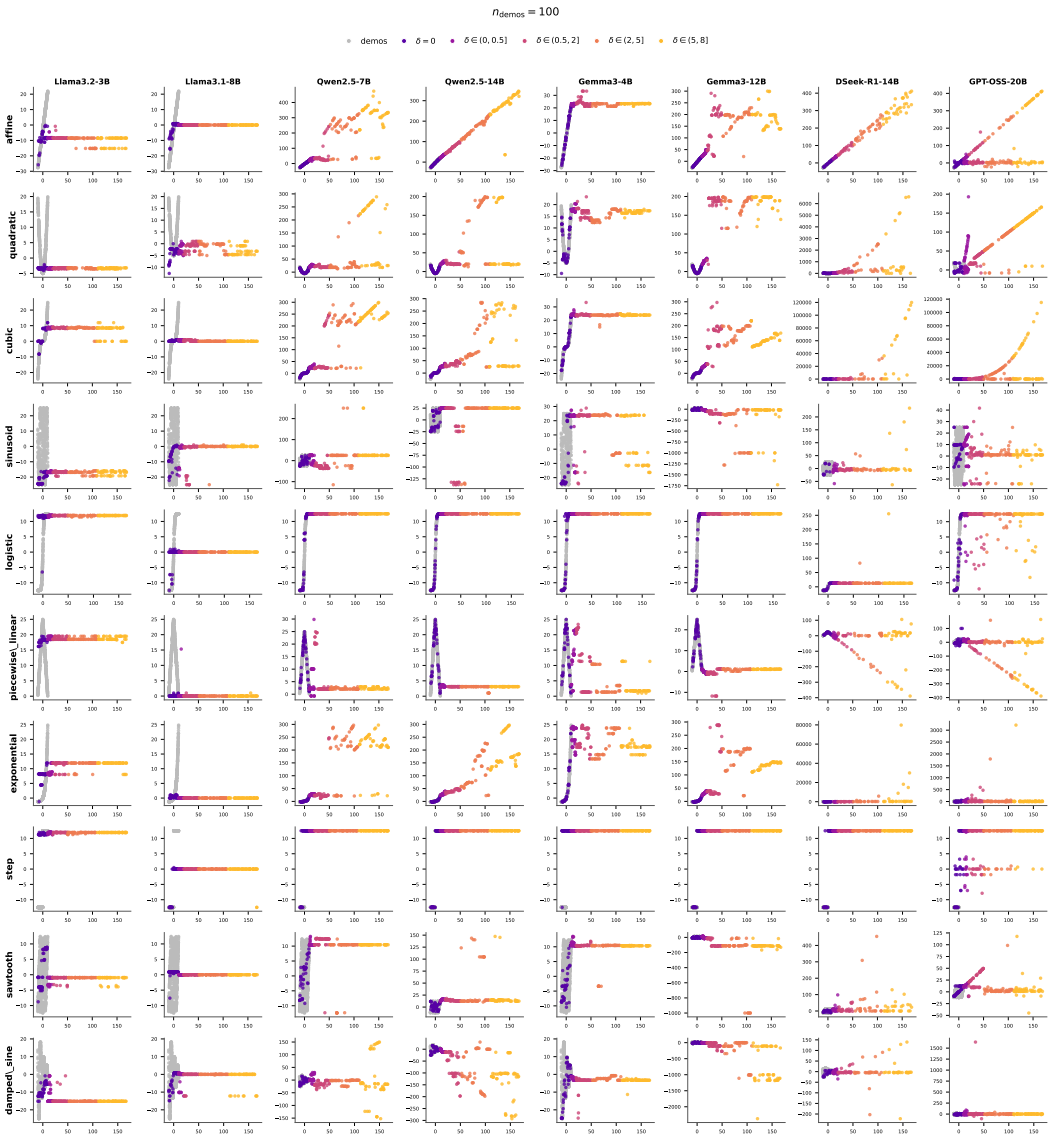

Figure 10: LLM predictions across (model, G) at $n_{\text{demos}} = 100$. Conventions as in Figure 6.

D RANSAC PEELING

This appendix provides the algorithmic detail behind (global modes via peeling). We describe the RANSAC primitive used to extract a single mode, the peeling loop that extracts multiple modes by sequential residualisation, the hyperparameters held fixed across all configurations, the full 30-configuration sweep, and the per-(model, G) decomposition for two contrasting configurations. Per-cell peel scatters and full per-config detail tables for all 30 configurations are in this section. In this section we use $B(X)$ to denote the number of fragments that we obtain by applying RANSAC over a surface X.

D.1 THE RANSAC PRIMITIVE

Given a pool of N candidate points $\{(x_i, y_i)\}_{i=1}^{N}$ and a hypothesis class $\mathcal{H}_k$ (HORIZONTAL, LINEAR, or POLY:d), RANSAC_ONE returns the largest subset of the pool that admits a single $\mathcal{H}_k$ fit within a residual tolerance τ. The pseudocode below mirrors the implementation in `ransac_peel.ransac_one`.

Algorithm 1 RANSAC_ONE$(\mathbf{x}, \mathbf{y}, \mathcal{H}_k, \tau_{\text{frac}}, T, \text{rng})$

Require: pool $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{N\times 2}$; hypothesis class $\mathcal{H}_k$ with sample-count $s \in \{1, 2, k+1\}$; threshold fraction τ_{frac}; iteration budget T; random generator `rng`.

$\tau \leftarrow \max(\tau_{\text{frac}}\,(\max \mathbf{y} - \min \mathbf{y}),\ 10^{-9})$
$C^\star \leftarrow -1,\ \mathcal{I}^\star \leftarrow \varnothing,\ \theta^\star \leftarrow \varnothing$
for $t = 1, \ldots, T$ **do**
  $S \leftarrow$ `rng.choice`$(N, s, \text{replace} = \text{False})$ ▷ minimal sample
  $\theta \leftarrow$ exact fit of $\mathcal{H}_k$ through S ▷ `np.polyfit`
  $\mathcal{I} \leftarrow \{i : |y_i - h_\theta(x_i)| \le \tau\}$
  if not IS_SANE$(\theta, \mathcal{I}, \tau)$ **then**
    continue ▷ rejects wiggly polynomials
  end if
  if $|\mathcal{I}| > C^\star$ **then**
    $C^\star, \mathcal{I}^\star, \theta^\star \leftarrow |\mathcal{I}|, \mathcal{I}, \theta$
  end if
end for
if $C^\star < N_{\text{min}}^{\text{in}}$ **then return** $\perp$
end if
$\theta^\star \leftarrow$ least-squares (or median, if horizontal) refit on $\mathcal{I}^\star$
$\mathcal{I}^\star \leftarrow \{i : |y_i - h_{\theta^\star}(x_i)| \le \tau\}$
return $(\mathcal{I}^\star, \theta^\star, S)$

The IS_SANE check guards against the classical RANSAC failure where a $\ge$ 2-degree polynomial is fit through three nearly-collinear points and produces a curve that, although matching the inliers, swings violently outside their convex hull. We reject any fit whose evaluated curve on the inlier x-range falls outside $[y_{\text{lo}} - 5\,y_{\text{span}},\ y_{\text{hi}} + 5\,y_{\text{span}}]$ where $y_{\text{span}} = \max(y_{\text{hi}} - y_{\text{lo}}, \tau) + \tau$.

D.2 THE PEELING LOOP

To extract multiple coexisting modes, RANSAC_ONE is applied sequentially to the residual pool. A configuration specifies an ordered list of *stages*, where each stage names a hypothesis class and an optional cap on how many modes that stage may extract before yielding to the next.

Algorithm 2 PEEL$(\mathbf{x}, \mathbf{y}, \text{stages}, \tau_{\text{frac}}, \text{rng})$

$R \leftarrow \{1, \ldots, N\}$, $\mathcal{M} \leftarrow [\,]$
for all $(\mathcal{H}_k, \text{cap}) \in$ stages **do**
  $j \leftarrow 0$
  while $|R| \geq N_{\text{min}}^{\text{rem}}$ **and** $|\mathcal{M}| < B_{\text{max}}$ **and** $(\text{cap} = \infty$ **or** $j < \text{cap})$ **do**
    $r \leftarrow$ RANSAC_ONE$(\mathbf{x}_R, \mathbf{y}_R, \mathcal{H}_k, \tau_{\text{frac}}, T, \text{rng})$
    if $r = \perp$ **then break**
    end if
    $(\mathcal{I}, \theta, S) \leftarrow r$
    Append mode $(\mathcal{H}_k, \theta, R[\mathcal{I}], R[S])$ to $\mathcal{M}$
    $R \leftarrow R \setminus R[\mathcal{I}]$
    $j \leftarrow j + 1$
  end while
end for
return $\mathcal{M}$

The number of modes returned for a given (model, G) cell, $|\mathcal{M}|$, is the LLM observation; running the same procedure on the ground-truth y_{test} pool of G yields the structural baseline $B(G)$. We report the deviation $\Delta = |\mathcal{M}| - B(G)$: positive Δ means the LLM's prediction surface is over-fragmented relative to the function class itself, negative means it has collapsed to fewer modes than the G requires.

D.3 Hyperparameters

Symbol	Value	Description
T	400	RANSAC iterations per call
$N_{\text{min}}^{\text{in}}$	10	minimum inliers to accept a mode
$N_{\text{min}}^{\text{rem}}$	10	minimum pool size to start a new round
B_{max}	30	cap on modes per cell
n_{demos}	100	demonstrations per cell
δ filter	$\delta > 0$	only extrapolation queries enter the pool
GLOBAL_SEED	0	root seed for SHA-256 cell-seed derivation

Table 4: Hyperparameters held fixed across all 30 peeling configurations. Per-cell determinism is enforced by deriving each cell's RANSAC seed from SHA256(GLOBAL_SEED | config_name | model | G), taking the first four bytes as a big-endian uint32. The threshold τ_{frac} is the only hyperparameter that varies across configurations.

D.4 The 30-configuration sweep

Configurations are organised into eight families, each parameterised by a list of threshold fractions τ_{frac}. Single-stage families use one hypothesis class with no cap; cascade families use a single-mode primary stage followed by an unrestricted secondary stage.

D.5 Master deviation table

Table 6 reports each (config, G) cell as B / Δ, where $B = B(G)$ is the baseline mode count obtained by peeling y_{test} under the same hypothesis (a structural property of $\mathcal{H}_G$, not of any LLM), and $\Delta = \text{mean}_{\text{models}}(\text{obs} - B)$ is the average deviation across the eight models. $\Delta \approx 0$ means the LLM matches the G's structural complexity under that hypothesis; $\Delta > 0$ means the LLM over-fragments (e.g. stepping a smooth function into multiple plateaus); $\Delta < 0$ means the LLM collapses into fewer modes than the function class admits (typically by failing to express periodic structure).

Two systematic patterns survive across the entire sweep, visible as column-Δ statistics in Table 6: over-fragmentation on monotone-growing Gs (EXPONENTIAL $\Delta = +2.6$, QUADRATIC $+2.0$), where the LLM expresses several quasi-flat or quasi-linear tracks where G admits a single fit; and under-expressivity on periodic Gs (SAWTOOTH $\Delta = -4.1$, SINUSOID -3.7), where the LLM collapses

#	Configuration	τ_{frac}	Stages
Single-stage horizontal (THR_DENSE: $1, 1.5, 3, 5, 6, 10\%$)			
1	H_010	1.0%	[horizontal]
2	H_015	1.5%	[horizontal]
3	H_030	3.0%	[horizontal]
4	H_050	5.0%	[horizontal]
5	H_060	6.0%	[horizontal]
6	H_100	10.0%	[horizontal]
Single-stage linear (THR_DENSE)			
7	L_010	1.0%	[linear]
8	L_015	1.5%	[linear]
9	L_030	3.0%	[linear]
10	L_050	5.0%	[linear]
11	L_060	6.0%	[linear]
12	L_100	10.0%	[linear]
Single-stage polynomial (THR_POLY: $1.5, 3, 5\%$)			
13	P2_015	1.5%	[poly:2]
14	P2_030	3.0%	[poly:2]
15	P2_050	5.0%	[poly:2]
16	P3_015	1.5%	[poly:3]
17	P3_030	3.0%	[poly:3]
18	P3_050	5.0%	[poly:3]
Line-then-horizontal cascades (THR_LH: $1.5, 3, 5, 6\%$)			
19	L_H_015	1.5%	[linear $\times 1$, horizontal]
20	L_H_030	3.0%	[linear $\times 1$, horizontal]
21	L_H_050	5.0%	[linear $\times 1$, horizontal]
22	L_H_060	6.0%	[linear $\times 1$, horizontal]
Poly-then-horizontal cascades (THR_PH: $1.5, 3, 5\%$)			
23	P2_H_015	1.5%	[poly:2 $\times 1$, horizontal]
24	P2_H_030	3.0%	[poly:2 $\times 1$, horizontal]
25	P2_H_050	5.0%	[poly:2 $\times 1$, horizontal]
26	P3_H_015	1.5%	[poly:3 $\times 1$, horizontal]
27	P3_H_030	3.0%	[poly:3 $\times 1$, horizontal]
28	P3_H_050	5.0%	[poly:3 $\times 1$, horizontal]
Poly-then-linear cascades (THR_PL: $3, 5\%$)			
29	P2_L_030	3.0%	[poly:2 $\times 1$, linear]
30	P2_L_050	5.0%	[poly:2 $\times 1$, linear]

Table 5: The 30 peeling configurations. Each is run on every (model, G) cell at $n_{\text{demos}} = 100$.

onto a small number of structured modes against $B \in \{8, 9\}$ for the ground-truth function. Both patterns are direct evidence that $\mathcal{F}_{\text{ICL}}$ is a multi-mode prediction surface whose structure does not faithfully track the hypothesis class of G.

D.6 PER-CELL VISUALISATION: H_015

Figure 11 shows the peel scatter for the H_015 configuration on every (model, G) cell. Pool points are coloured by the horizontal mode they were assigned to; grey points were not absorbed by any mode. The simplest non-trivial configuration in the sweep — horizontal-only at $\tau_{\text{frac}} = 1.5\%$ — decomposes the prediction surface into a sequence of flat bands. Table 7 lists the per-cell deviation Δ from the G baseline. Two patterns dominate: $\Delta > 0$ where the LLM fragments a smooth function into several plateaus (most pronounced on EXPONENTIAL, where every model adds 1–8 extra plateaus on top of the $B = 1$ baseline), and $\Delta < 0$ on PIECEWISE_LINEAR where several models collapse the two slopes of the ground truth into a single horizontal mode.



Figure 11: Peel scatter for H_015 (horizontal-only, $\tau_{\text{frac}} = 1.5\%$). Rows: LLMs; columns: data-generating processes. Each panel plots the test-query pool $(x_j^{\text{test}}, y_j^{\text{LLM}})$ at $n_{\text{demos}} = 100$, $\delta > 0$; coloured points are inliers of the corresponding peeled mode; grey points were not assigned to any mode. Panel titles report the number of modes recovered.

Table 6: Modes relative to the G baseline. Each cell: $\mathbf{B}$ / Δ, where $\mathbf{B}$ is the number of modes found by peeling the ground-truth y_{test} under the stated hypothesis and $\Delta = \text{mean}_{\text{models}}(\text{obs} - B)$ is the average LLM deviation from that baseline. $\Delta \approx 0$: LLM matches G's structural complexity. $\Delta > 0$: over-fragmented (e.g. plateau-stepping). $\Delta < 0$: collapsed / under-expressive. The row Δ column averages Δ over the ten Gs; the col Δ row averages Δ over the thirty configurations.

config	affine	cubic	damped_sine	exponential	logistic	piecewise_linear	quadratic	sawtooth	sinusoid	step	*row* Δ
h_010	3/+2.5	3/+2.4	2/+2.4	1/+5.1	1/+1.2	3/+1.2	3/+2.8	0/+4.0	3/+2.2	1/+1.2	**+2.5**
h_015	4/+1.8	3/+1.8	1/+3.4	1/+5.2	1/+1.9	6/-1.4	4/+2.1	3/+1.0	6/-0.4	1/+1.2	**+1.7**
h_030	7/-1.2	4/+0.8	3/+1.0	1/+4.2	1/+1.8	8/-3.6	5/+1.5	11/-7.4	8/-3.0	1/+1.2	**-0.5**
h_050	8/-3.1	4/+0.5	3/+1.1	1/+4.0	2/+0.9	8/-3.8	6/-1.0	9/-5.6	9/-4.8	1/+1.1	**-1.1**
h_060	7/-2.1	4/-0.1	3/+0.6	1/+3.5	1/+1.8	7/-2.8	6/-1.2	8/-4.9	8/-3.8	1/+1.1	**-0.8**
h_100	6/-2.4	4/-0.6	4/-1.2	1/+2.8	1/+1.4	6/-2.6	4/-0.6	6/-3.2	6/-2.2	1/+1.1	**-0.8**
l_010	1/+2.6	4/+0.1	4/+0.2	1/+3.2	2/-0.2	1/+3.4	4/+1.2	5/-1.4	5/-0.5	1/+1.2	**+1.0**
l_015	1/+2.2	4/-0.2	3/+1.2	1/+3.0	2/+0.2	1/+3.1	3/+1.8	5/-1.6	8/-3.6	1/+1.2	**+0.7**
l_030	1/+1.8	3/+0.1	5/-1.4	1/+2.5	2/+0.2	1/+2.6	2/+1.9	9/-6.0	9/-5.1	1/+1.2	**-0.2**
l_050	1/+1.4	2/+0.9	4/-1.2	1/+1.9	2/+0.0	1/+2.0	2/+1.2	9/-6.2	9/-5.6	1/+1.1	**-0.5**
l_060	1/+1.4	2/+1.0	3/-0.2	1/+1.9	2/+0.1	1/+2.0	2/+1.0	7/-4.5	8/-4.8	1/+1.1	**-0.1**
l_100	1/+1.1	2/+0.4	3/-0.6	1/+1.5	1/+0.9	1/+1.6	1/+1.5	6/-3.9	6/-3.2	1/+1.0	**+0.0**
p2_015	1/+2.2	2/+1.9	4/+0.0	1/+3.0	2/+0.5	1/+3.1	1/+3.4	8/-4.4	6/-1.8	1/+1.2	**+0.9**
p2_030	1/+1.8	1/+2.1	4/-0.8	2/+1.2	2/-0.1	1/+2.2	1/+2.6	9/-6.1	9/-5.5	1/+1.1	**-0.1**
p2_050	1/+1.5	1/+2.0	4/-1.1	2/+1.0	2/-0.1	1/+1.8	1/+1.9	8/-5.4	7/-3.9	1/+1.1	**-0.1**
p3_015	1/+2.2	1/+2.6	5/-1.1	2/+2.1	2/+0.4	1/+2.9	1/+3.4	8/-4.8	9/-4.9	1/+1.2	**+0.4**
p3_030	1/+1.6	1/+1.9	4/-0.6	2/+1.1	2/+0.0	1/+2.1	1/+2.5	9/-6.1	10/-6.2	1/+1.1	**-0.3**
p3_050	1/+1.4	1/+1.6	4/-1.2	2/+0.9	2/-0.1	1/+1.8	1/+1.8	8/-5.5	8/-4.9	1/+1.1	**-0.3**
l_h_015	1/+2.0	1/+2.5	1/+2.8	1/+3.2	1/+0.9	1/+2.2	1/+3.9	2/+1.5	5/-0.4	1/+1.2	**+2.0**
l_h_030	1/+1.6	2/+1.2	3/+0.2	1/+2.5	1/+0.8	1/+1.9	2/+1.9	10/-6.9	9/-4.9	1/+1.1	**-0.1**
l_h_050	1/+1.5	2/+0.6	3/+0.1	1/+1.8	1/+0.8	1/+2.0	2/+1.4	9/-6.2	8/-4.8	1/+1.1	**-0.2**
l_h_060	1/+1.6	2/+0.8	3/+0.0	1/+1.6	1/+0.8	1/+2.4	2/+1.1	8/-5.2	8/-4.5	1/+1.1	**-0.0**
p2_h_015	1/+2.0	1/+2.5	1/+2.8	1/+3.2	1/+0.9	1/+2.5	1/+3.5	2/+1.4	5/-0.4	1/+1.2	**+2.0**
p2_h_030	1/+1.9	1/+2.2	3/+0.0	1/+2.1	1/+0.8	1/+2.1	1/+3.2	10/-6.9	8/-3.8	1/+1.1	**+0.3**
p2_h_050	1/+1.8	1/+1.9	3/+0.2	1/+2.0	1/+0.8	1/+1.9	1/+2.4	8/-5.2	10/-6.5	1/+1.1	**+0.0**
p3_h_015	1/+2.0	1/+2.9	1/+2.4	1/+3.5	1/+0.9	1/+2.4	1/+3.6	2/+1.2	7/-2.5	1/+1.2	**+1.8**
p3_h_030	1/+1.6	1/+2.0	3/+0.0	1/+2.2	1/+0.8	1/+2.1	1/+3.4	10/-6.9	8/-3.9	1/+1.1	**+0.2**
p3_h_050	1/+1.5	1/+2.0	2/+1.1	1/+2.0	1/+0.8	1/+1.8	1/+2.5	9/-6.2	10/-6.6	1/+1.1	**-0.0**
p2_l_030	1/+1.9	1/+2.1	5/-1.6	1/+2.2	2/+0.1	1/+2.5	1/+2.8	8/-5.0	9/-5.1	1/+1.1	**+0.1**
p2_l_050	1/+1.4	1/+1.9	5/-2.2	1/+2.1	2/+0.0	1/+1.9	1/+2.2	8/-5.1	9/-5.6	1/+1.1	**-0.2**

Table 7: Deviation from G baseline for RANSAC peel (h_015) — horizontal @ 1.5%. Cells are obs $- B$; the *baseline* row lists B (modes found by peeling y_{test}).

model	affine	cubic	damped_sine	exponential	logistic	piecewise_linear	quadratic	sawtooth	sinusoid	step	mean Δ
baseline (B)	4	3	1	1	1	6	4	3	6	1	–
Qwen2.5-14B	+4	+3	+5	+8	+1	-4	+3	+2	-1	+1	+2.2
Qwen2.5-7B	+4	+5	+5	+6	+1	-3	+1	+0	+0	+1	+2.0
DSeek-R1-14B	+3	+4	+5	+8	+8	+4	+3	+3	+3	+0	+4.1
Gemma-3-12B	+6	+4	+3	+8	+1	-3	+3	+0	-1	+1	+2.2
Gemma-3-4B	-1	+0	+3	+8	+1	+3	+1	+0	+0	+0	+1.5
Llama-3.1-8B	-2	+0	+4	+1	+0	-5	+3	-1	-3	+1	-0.2
Llama-3.2-3B	-1	+0	+2	+2	+1	+0	-2	+0	-1	+3	+0.4
GPT-OSS-20B	+1	-2	+0	+1	+2	-3	+5	+4	+0	+3	+1.1
mean Δ	+1.8	+1.8	+3.4	+5.2	+1.9	-1.4	+2.1	+1.0	-0.4	+1.2	+1.7

D.7 PER-CELL VISUALISATION: P2_L_030

Figure 12 shows the peel scatter for the cascade P2_L_030 (one degree-2 polynomial peeled first, then linear bands at $\tau_{\text{frac}} = 3\%$). The richer hypothesis class absorbs more of the surface into structured modes, dropping the global mean Δ to $+0.1$, but the residual structure shown in Table 8 is informative. Positive Δ persists on CUBIC, EXPONENTIAL, PIECEWISE_LINEAR, and QUADRATIC (LLMs over-fragment smooth growth into multiple linear tracks), and strongly negative Δ on SAWTOOTH and

SINUSOID indicates that LLMs fail to express the periodic structure these Gs require ($B = 8$ and $B = 9$ respectively); they collapse onto 3–5 modes instead.

Table 8: Deviation from G baseline for RANSAC peel (`p2_l_030`) — poly2 then linear @ 3%.

model	affine	cubic	damped_sine	exponential	logistic	piecewise_linear	quadratic	sawtooth	sinusoid	step	mean Δ
baseline (B)	1	1	5	1	2	1	1	8	9	1	–
Qwen2.5-14B	+0	+3	-1	+2	+1	+2	+3	-5	-5	+1	+0.1
Qwen2.5-7B	+5	+3	-1	+4	+1	+3	+2	-4	-6	+1	+0.8
DSeek-R1-14B	+0	+0	-3	+0	-1	+0	+0	-5	-7	+0	-1.6
Gemma-3-12B	+4	+4	-1	+4	+1	+3	+3	-5	-6	+1	+0.8
Gemma-3-4B	+2	+2	-1	+5	+0	+6	+4	-4	-5	+0	+0.9
Llama-3.1-8B	+1	+2	-1	+1	-1	+0	+6	-6	-5	+1	-0.2
Llama-3.2-3B	+2	+1	-1	+2	-1	+4	+1	-5	-4	+2	+0.1
GPT-OSS-20B	+1	+2	-4	+0	+1	+2	+3	-6	-3	+3	-0.1
mean Δ	+1.9	+2.1	-1.6	+2.2	+0.1	+2.5	+2.8	-5.0	-5.1	+1.1	+0.1

E ACTIVATION STEERING

The steering experiment described the in main text tests whether the modes recovered by peeling correspond to distinct directions in the LLM's residual stream. If they do, a forward-hook intervention that adds a small multiple of a mode's representative direction at one mid-network layer should reroute the prediction toward that mode on held-out prompts; subtraction should push it away. This appendix gives the algorithmic detail of the pipeline that produces those steering vectors and applies them.

E.1 PIPELINE OVERVIEW

For a fixed (model, G, n_{demos}) cell with K peeled modes, the pipeline runs in four stages.

1. **Donor collection.** Apply RANSAC peeling (previous Appendix) to the cell's pool of test queries to identify K modes. For each mode, split its inliers into a *donor* set (used to estimate the mode's representative direction) and a disjoint *holdout* set (used to evaluate steering).
2. **Activation capture.** For every donor and holdout prompt, record the residual-stream activation at the last input token at L evenly-spaced layers across the network.
3. **Layer ranking.** For every layer, score the separability of the K donor groups by Cohen's d along the principal donor-mean axis. Define the steering vector at each layer as the inter-mode mean difference; report the layer with the highest $|d|$ as the steering target.
4. **Steering.** At the chosen layer, register a forward hook that adds $\alpha\,\hat{v}$ (where $\hat{v}$ is a unit-normalised inter-mode direction and α is in units of $\|v\|$) to the residual stream at the last input-token position. Sweep $\alpha \in [\alpha_{\min}, \alpha_{\max}]$, decode predictions for every held-out prompt, classify each prediction by closest mode, and report mode-flip rates as a function of α.

E.2 DONOR COLLECTION

Pool construction is identical to peeling: concatenate the test-query predictions $(x^{\text{test}}, y^{\text{LLM}})$ across the five δ-bins of the cell, drop $\delta = 0$ (interpolation queries), and drop any unparsed predictions. RANSAC peeling under a two-hypothesis cascade (HORIZONTAL primary, LINEAR residual; threshold $\tau_{\text{frac}} = 3\%$, $T = 400$ iterations, $N^{\text{in}}_{\text{min}} = 10$) produces $K = 2$ inlier sets in our experiments; richer cascades produce more (e.g., $K = 3$ on QWEN-2.5-7B/AFFINE, where peeling recovers a true linear mode and two distinct constant collapse modes; main-text Figure 3). For each mode we sample disjointly n_{donor} donor points and n_{holdout} holdout points under a fixed seed; the remaining inliers are

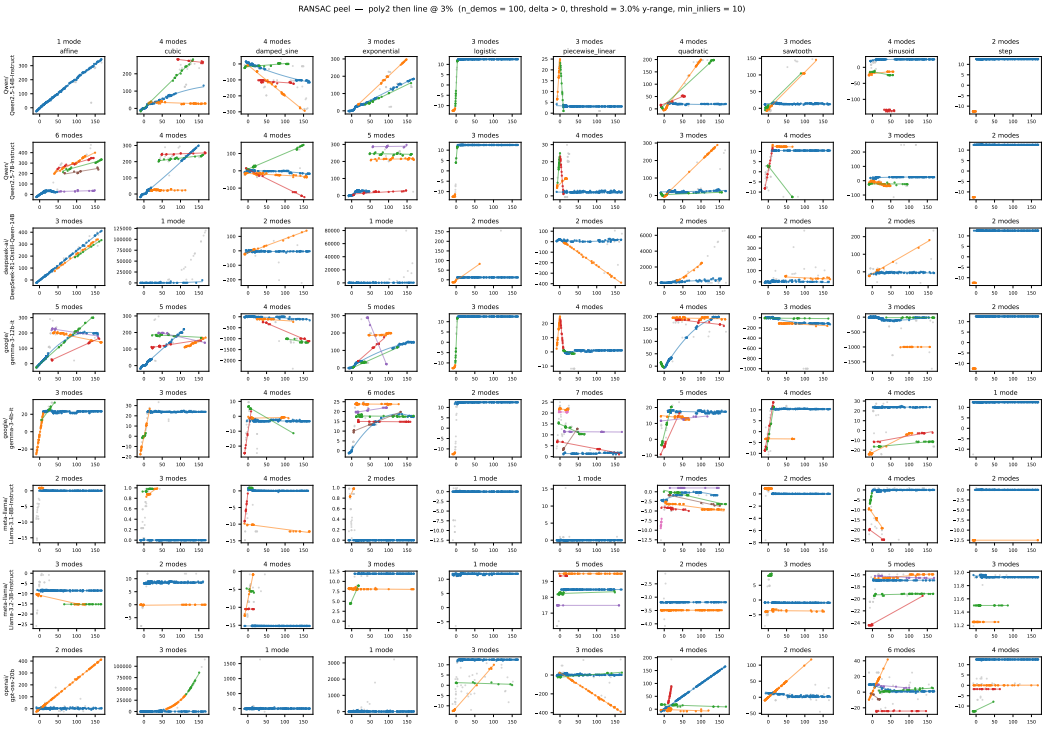

Figure 12: Peel scatter for P2_L_030 (poly2 then linear, $\tau_{\text{frac}} = 3\%$). Rows: LLMs; columns: data-generating processes. Same conventions as Figure 11; the first mode in each panel is a degree-2 polynomial, subsequent modes are linear bands.

pooled and not used. A single layout used throughout the experiment is $n_{\text{donor}} = n_{\text{holdout}} = 20$ per mode.

E.3 Activation capture

For a transformer with N decoder blocks, we capture residual-stream activations at $L = 7$ evenly-spaced indices. For each prompt the activation is taken at the last input-token position (`<|im_start|>assistant` prefix, before the model emits any output token), giving a tensor of shape $(n_{\text{donor}} + n_{\text{holdout}}) \cdot K \times L \times d$, where d is the model's residual dimension. Decoding is greedy (`do_sample=False`, `temperature=0`) and special-token handling matches the one given in base output parsing in Appendix B.

E.4 Layer ranking

For each captured layer $\ell \in \{1, \ldots, L\}$ we compute the inter-mode mean direction $v_\ell = \mu_\ell^{(1)} - \mu_\ell^{(0)} \in \mathbb{R}^d$ (extended to multi-mode pairs by taking each pair separately). Project all donor activations at layer ℓ onto the unit vector $\hat{v}_\ell$ and compute the per-layer Cohen's d between the two projected distributions:

$$d_\ell \;=\; \frac{\bar{p}_\ell^{(1)} - \bar{p}_\ell^{(0)}}{\sqrt{\frac{1}{2}\big(\sigma_\ell^{2,(0)} + \sigma_\ell^{2,(1)}\big)}}, \qquad p_{\ell,n}^{(k)} \;=\; \langle a_{\ell,n}^{(k)},\, \hat{v}_\ell \rangle.$$

We rank layers by $|d_\ell|$ and select the top-ranked layer $\ell^\star$ as the steering target. Two complementary diagnostics are reported in the supplementary material: $\|\mu_\ell^{(1)} - \mu_\ell^{(0)}\|$ divided by the pooled norm $\|\sigma\|$, and the within-donor binary-classification accuracy of $\text{sign}(\langle a, \hat{v}_\ell \rangle - \tau_\ell)$ where $\tau_\ell = \frac{1}{2}(\bar{p}_\ell^{(0)} + \bar{p}_\ell^{(1)})$. The three diagnostics agree on the top layer in every cell we tested.

E.5 Steering

At the chosen layer $\ell^\star$ we register a forward hook on the $\ell^\star$-th decoder block. Let $h_t \in \mathbb{R}^d$ be the residual stream at token position t, and let T be the position of the last input token (the same position used for activation capture). The hook applies

$$h'_T \;=\; h_T \;+\; \alpha \cdot \|v_{\ell^\star}\| \cdot \hat{v}_{\ell^\star}, \qquad \alpha \in \{-4, -2, -1, -0.5, 0, +0.5, +1, +2, +4\},$$

leaving all other token positions untouched. Sign of α encodes direction: $\alpha > 0$ pushes the residual stream from mode 0 toward mode 1 (positive injection / induction); $\alpha < 0$ pushes it away from mode 1 toward mode 0 (negative injection / removal). Scaling by $\|v_{\ell^\star}\|$ makes α dimensionless — $\alpha = 1$ corresponds to a perturbation of the same Euclidean magnitude as the donor-group mean difference itself.

For every held-out prompt and every α, the model is queried under the active hook, the completion is parsed as in Appendix B, and the predicted scalar y^{LLM} is classified:

$$\text{label}(y^{\text{LLM}}) \;=\; \begin{cases} 0 & \text{if } |y^{\text{LLM}} - h_0(x^{\text{test}})| < b \text{ and closer to mode 0} \\ 1 & \text{if } |y^{\text{LLM}} - h_1(x^{\text{test}})| < b \text{ and closer to mode 1} \\ \perp & \text{otherwise} \end{cases}$$

where h_0, h_1 are the peeled-mode functions of §E.2 and the band $b = \max\big(\tau,\ \frac{1}{2}\,|h_0(x^{\text{test}}) - h_1(x^{\text{test}})|\big)$ is the larger of an absolute tolerance τ and half the inter-mode gap at x^{test}. The two-band construction prevents predictions in the overlap region of similar-mode queries from being miscalled as $\perp$.

For every mode pair (j, k) and every α, the *induction rate* (positive direction) is the fraction of holdout queries that started in mode j and were classified as mode k at $\alpha > 0$; the *removal rate* (negative direction) is the fraction of holdout queries that started in mode k and were classified outside mode k at $\alpha < 0$. Reading both directions is essential: the asymmetry between induction and removal distinguishes true attractors (vector acts as a basin of attraction in both directions) from re-routings (a steering vector that lifts a query out of one mode but disperses rather than channels into the target).

E.6 GENERALISATION TO $K > 2$ MODES

The pipeline above extends naturally to K peeled modes: instead of a single inter-mode vector v_ℓ, we compute $\binom{K}{2}$ pairwise vectors $v_\ell^{(j,k)} = \mu_\ell^{(k)} - \mu_\ell^{(j)}$. For QWEN-2.5-7B/AFFINE/d=100, peeling recovers $K = 3$ modes (r_1 true linear, r_2 collapse to demonstration mean, r_3 collapse to a low-valued constant; main-text Figure 3). We construct three pairwise vectors $v^{(1,2)}, v^{(1,3)}, v^{(2,3)}$ at the layer of highest mean $|d|$ (layer 26 for Qwen-2.5-7B). Positive and negative steering along each vector reroutes the prediction in the predicted directions and at the predicted target rates; PCA of the residual stream at layer 26 shows three visibly separated sub-populations (Figure 3d).

E.7 HYPERPARAMETERS

Parameter	Value	Description
n_{donor}	20	donor prompts per peeled mode
n_{holdout}	20	disjoint holdout prompts per mode
RANSAC stages	[HORIZONTAL $\times 1$, LINEAR]	two-mode peeling
τ_{frac}	3%	RANSAC tolerance (fraction of pool y-range)
T (RANSAC)	400	RANSAC iterations per peel
L	7	layers captured (evenly spaced)
Token position	last	last input-token before generation
Layer-ranking metric	Cohen's d	along $\hat{v}_\ell = (\mu_\ell^{(1)} - \mu_\ell^{(0)})/\|\cdot\|$
α sweep	$\{\pm 4, \pm 2, \pm 1, \pm 0.5, 0\}$	in units of $\|v_{\ell^\star}\|$
Decoding	greedy ($T = 0$)	matches Appendix B
Classification band b	$\max(\tau, \frac{1}{2}\Delta_y)$	$\Delta_y = \|h_0 - h_1\|$ at x^{test}

Table 9: Hyperparameters for the steering pipeline. The same settings are used for every (model, G) cell unless noted; the only per-cell decision is the choice of steering layer $\ell^\star$, which is selected automatically by Cohen's d.

E.8 REPRODUCIBILITY

The pipeline is implemented as four standalone scripts (`steering_new/run/1_collect_donors.py`, `2_capture_activations.py`, `3_analyze_layers.py`, `4_steer.py`); each writes a single `.npz` sidecar that the next script consumes. Donor selection, activation capture, and steering are all seeded; the RANSAC seed for donor collection is SHA256(GLOBAL_SEED | config_name | model | $\mathcal{G}$) as in previous Appendix section. Re-running the pipeline from scratch on the same hardware reproduces the steering curves bit-for-bit.

E.9 STEERING ACROSS MODELS

We repeated the steering protocol above on four models, using disjoint donor and holdout queries, layer selection by Cohen's d, and the same mode-band classifier. Table 10 reports induction into a target peeled mode under positive steering and removal from the original mode under negative steering. Induction ranges from 79% to 89%, and removal from 63% to 77%; the model-wise means are 84% and 69%. Thus, peeled modes correspond to steerable residual-stream directions in each tested model.

F MODELS AND APPROXIMATION ERRORS

Figure 13 provides a baseline view of predictive accuracy by reporting $\log_{10}(\text{RMSE})$ across the 8×10 model–$\mathcal{G}$ grid, stratified by distance bin δ. In the interpolation regime ($\delta = 0$), errors are relatively compressed across models, with no model uniformly dominating across choice of $\mathcal{G}$. As δ increases, the heatmaps develop a strong column-wise structure: error varies more by target $\mathcal{G}$ than by model. This is most visible in the farthest bin, where functions like exponential which has 'larger values' in true range compared to demonstrations. This produces uniformly catastrophic error across

Model	Peeled modes	Induction	Removal
Qwen-2.5-7B	3	83%	63%
Gemma-3-4B	2	86%	71%
Gemma-3-12B	2	89%	77%
Llama-3.1-8B	2	79%	65%
Mean	2.25	84%	69%

Table 10: Held-out steering results. Induction and removal follow the definitions in §E.5; means are unweighted across models.

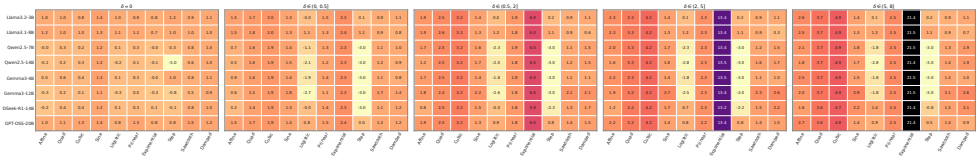

Figure 13: Geometric-mean RMSE across sparsity levels for each (model, $\mathcal{G}$) cell, stratified by normalized extrapolation distance δ. Cell annotations and color encode $\log_{10}(\mathrm{RMSE})$; shared scale across all five panels. Negative numbers indicate near zero errors.

models, while bounded $\mathcal{G}$s such as logistic gives smaller errors. For polynomial $\mathcal{G}$, error grows in an orderly way with both distance and polynomial degree, rather than degrading arbitrarily.

Structured growth of error with δ suggests that models are not predicting using the true data generating mechanism, at least when queries far away from the demonstration. Though this does not say anything about multiple modes, but the orderliness itself is informative: a model executing the true data generation mechanism would not degrade in a δ-graded, manner. The smooth, structured growth therefore implies the predictive mode itself shifts from true function with query location — an observation which could be explained as noise in prediction with decreasing similarity of query from demonstrations.

G Pointwise attribution-profile grids

For each of the eight LLMs, the figures below show the attribution profile across all 50 $(\mathcal{G}, \delta)$ pairs (10 $\mathcal{G}$s $\times$ 5 δ-bins), with sparsities pooled within each panel. Within every panel the bars over the five analytic shapes (Linear, Polynomial, Tree, Kernel, Periodic) are stacked into a solid lower segment for global-fit mass and a hatched upper segment for local-fit mass; `KNN_Regression` is folded into the Kernel/local segment as a uniform-kernel local smoother. These per-model grids supply the 400 panels.

G.1 Tie-aware analysis of pointwise attribution

Several classical predictors can give nearly identical answers at a query, making a hard argmin attribution sensitive to small differences in their predictions. We therefore measure the separation between the best-matching families and repeat the attribution with shared credit for near-ties. For a predictor c_a^D fitted to demonstrations D, let $\mathcal{A}_f$ index the predictors in family f. Its divergence from the LLM prediction $\hat{y}^q$ is

$$\Delta_f(x^q) = \min_{a \in \mathcal{A}_f} |c_a^D(x^q) - \hat{y}^q|, \qquad M(x^q) = \Delta_{(2)}(x^q) - \Delta_{(1)}(x^q),$$

where $\Delta_{(1)}$ and $\Delta_{(2)}$ are the smallest two family divergences. Within each $(\mathrm{model}, G, n, \delta\text{-bin})$ cell, we set τ_{cell} to the median $\Delta_{(1)}$ across queries, floored at 10^{-9}. This is an empirical resolution scale given by the typical mismatch to the closest predictor. Queries with $M(x^q) \le \tau_{\mathrm{cell}}$ are treated as ambiguous at this resolution.

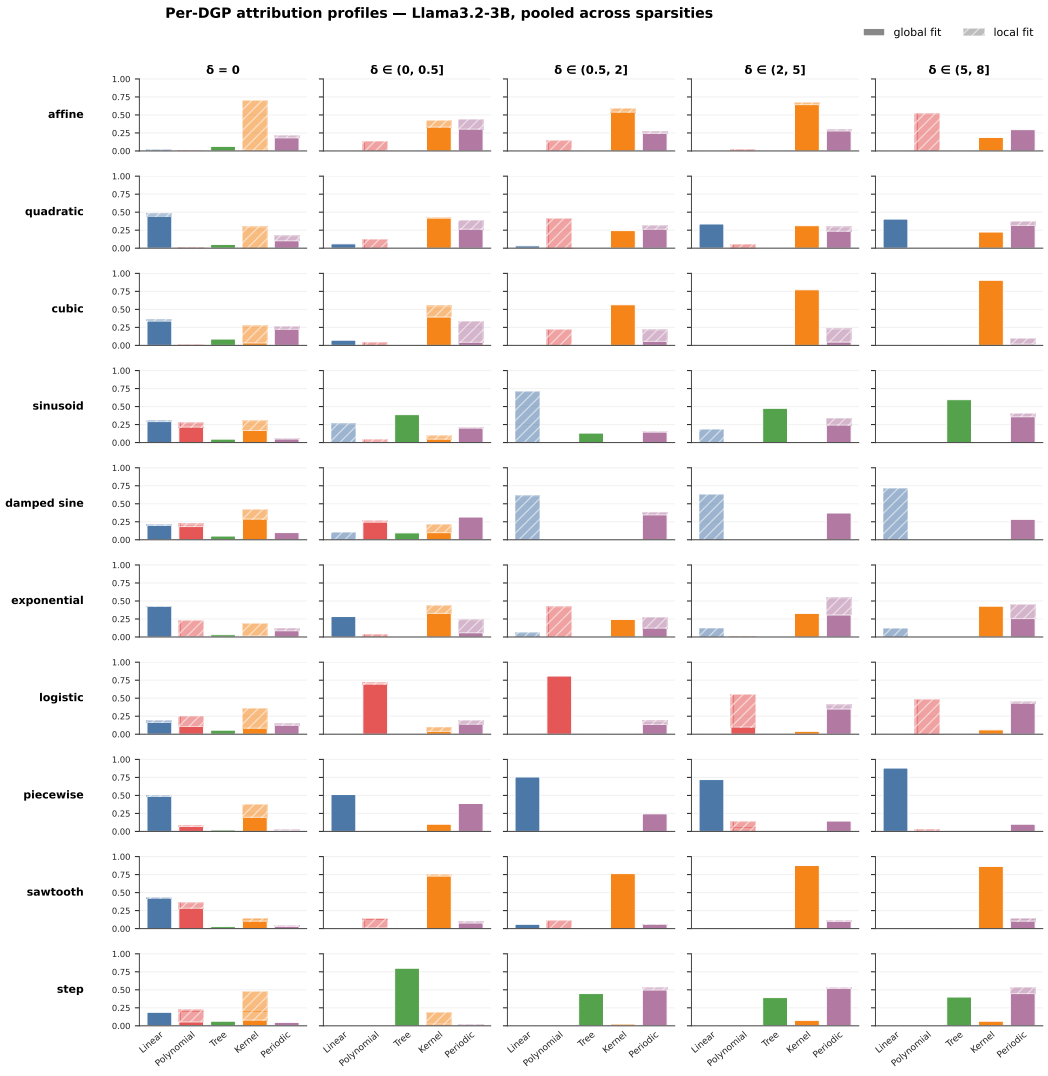

Figure 14: Attribution profiles for *Llama3.2-3B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.



Figure 15: Attribution profiles for *Llama3.1-8B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

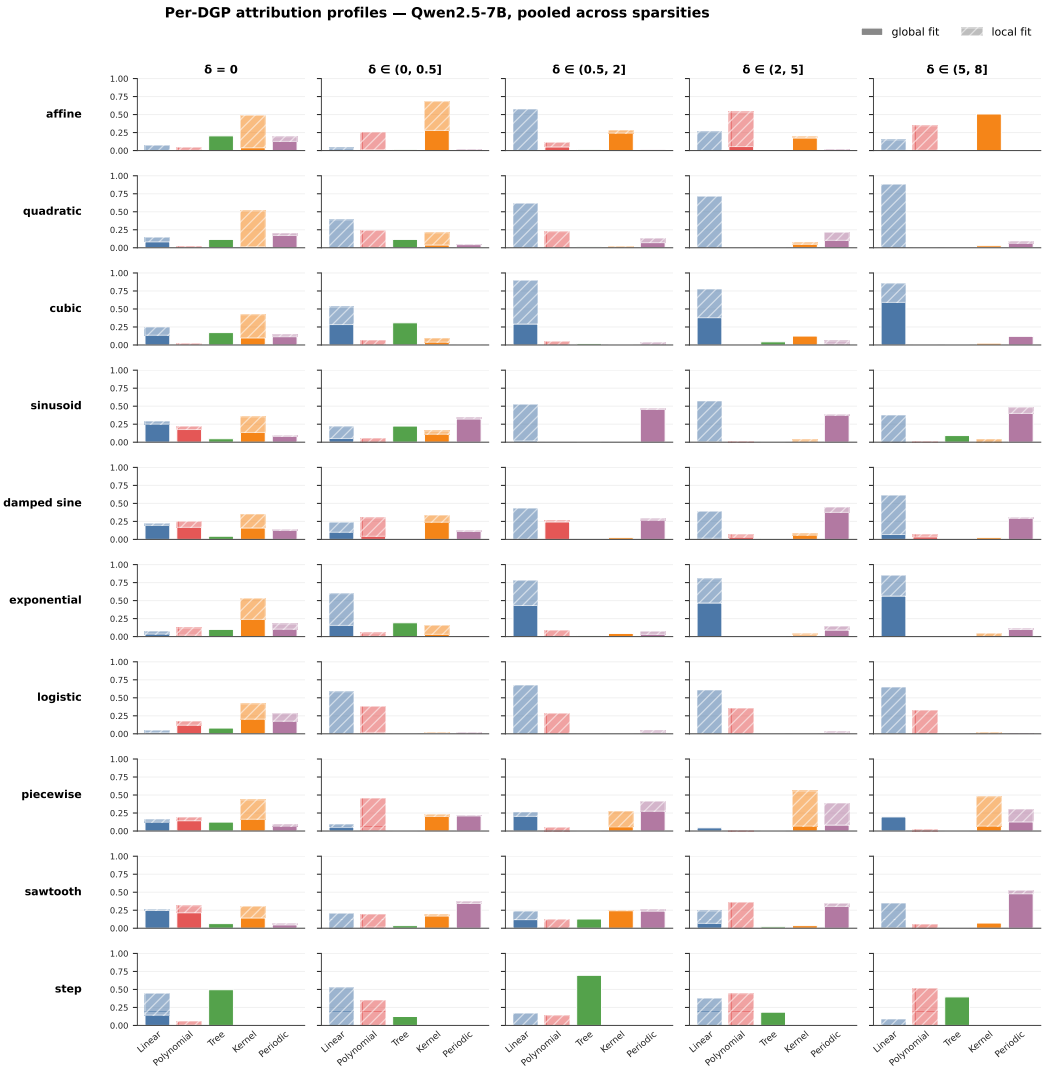

Figure 16: Attribution profiles for *Qwen2.5-7B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

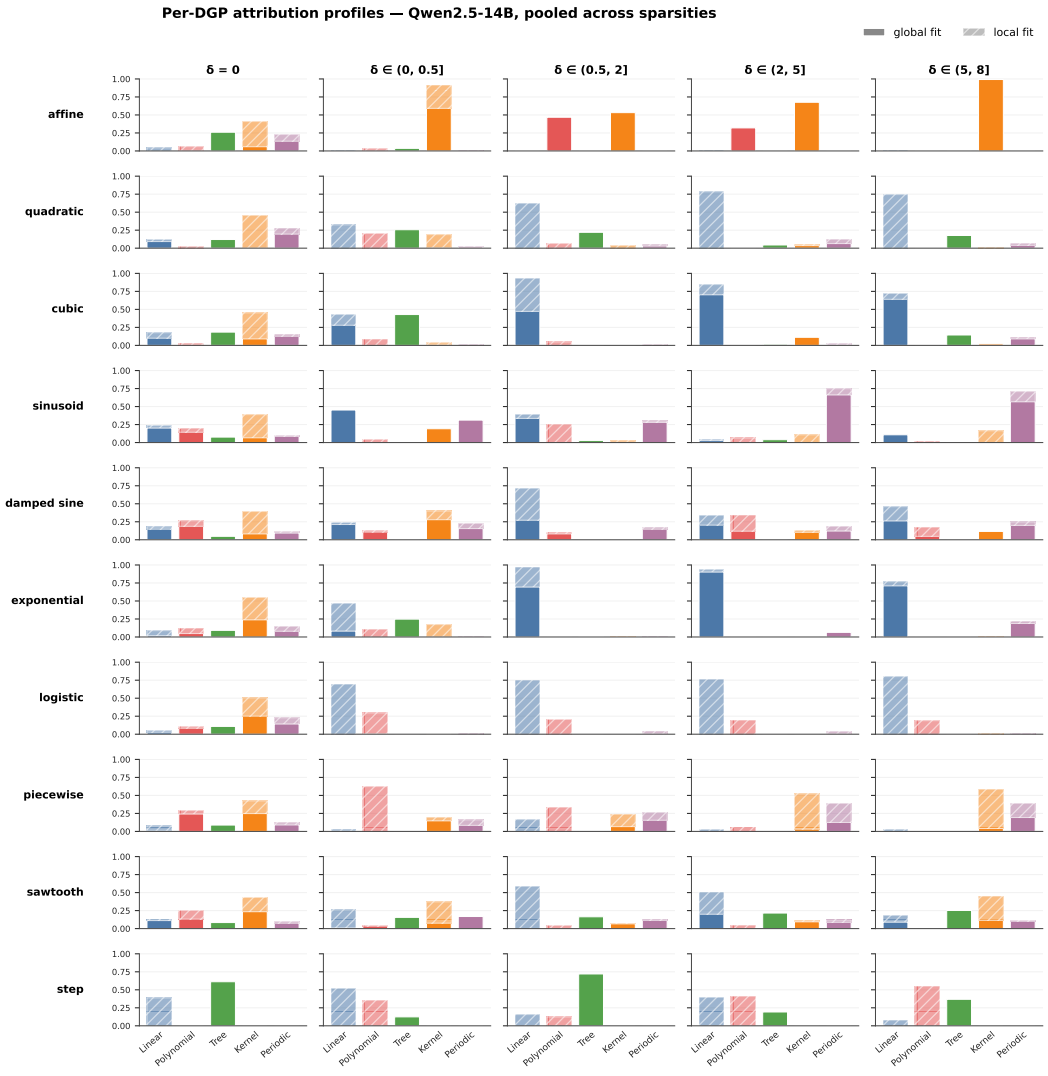

Figure 17: Attribution profiles for *Qwen2.5-14B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

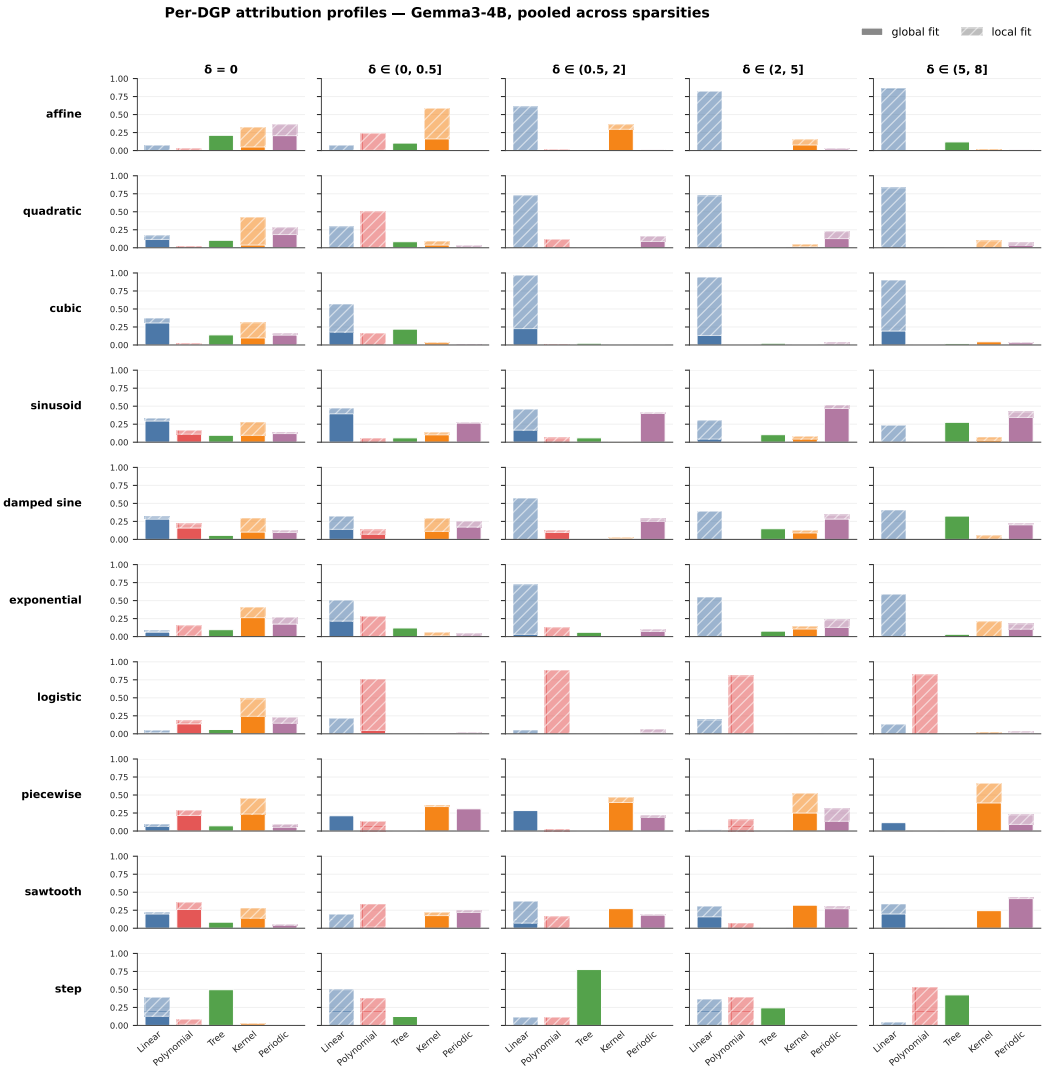

Figure 18: Attribution profiles for *Gemma3-4B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

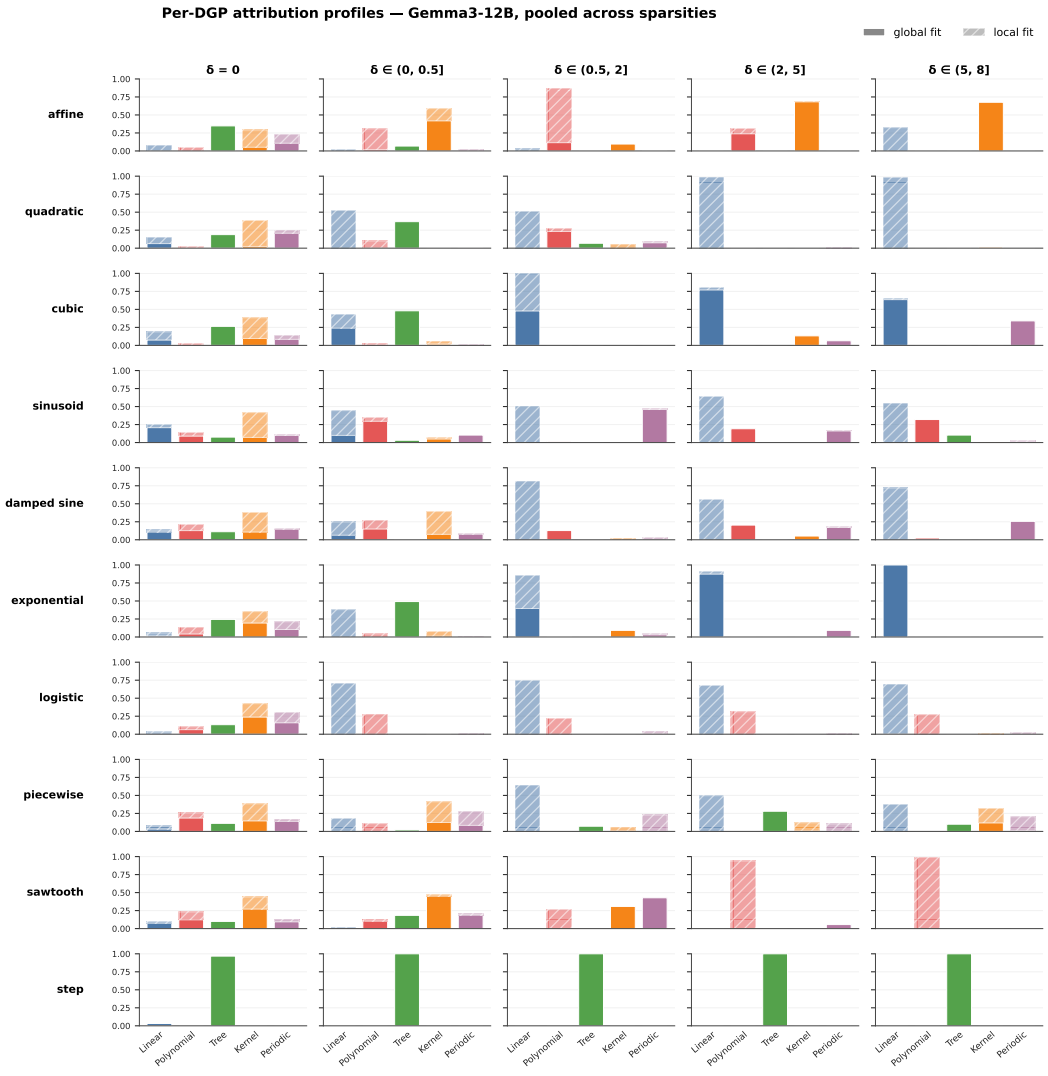

Figure 19: Attribution profiles for *Gemma3-12B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

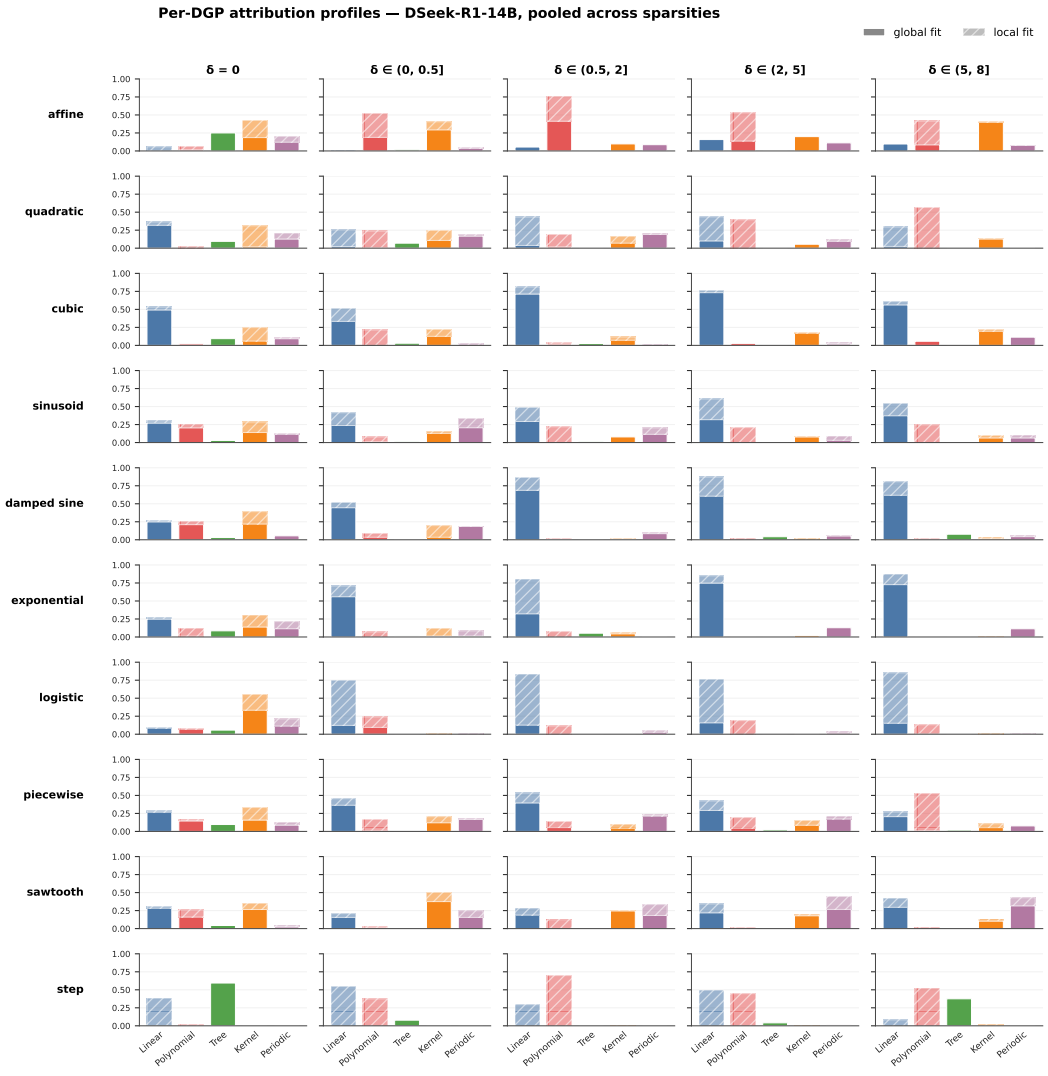

Figure 20: Attribution profiles for *DeepSeek-R1-Distill-Qwen-14B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

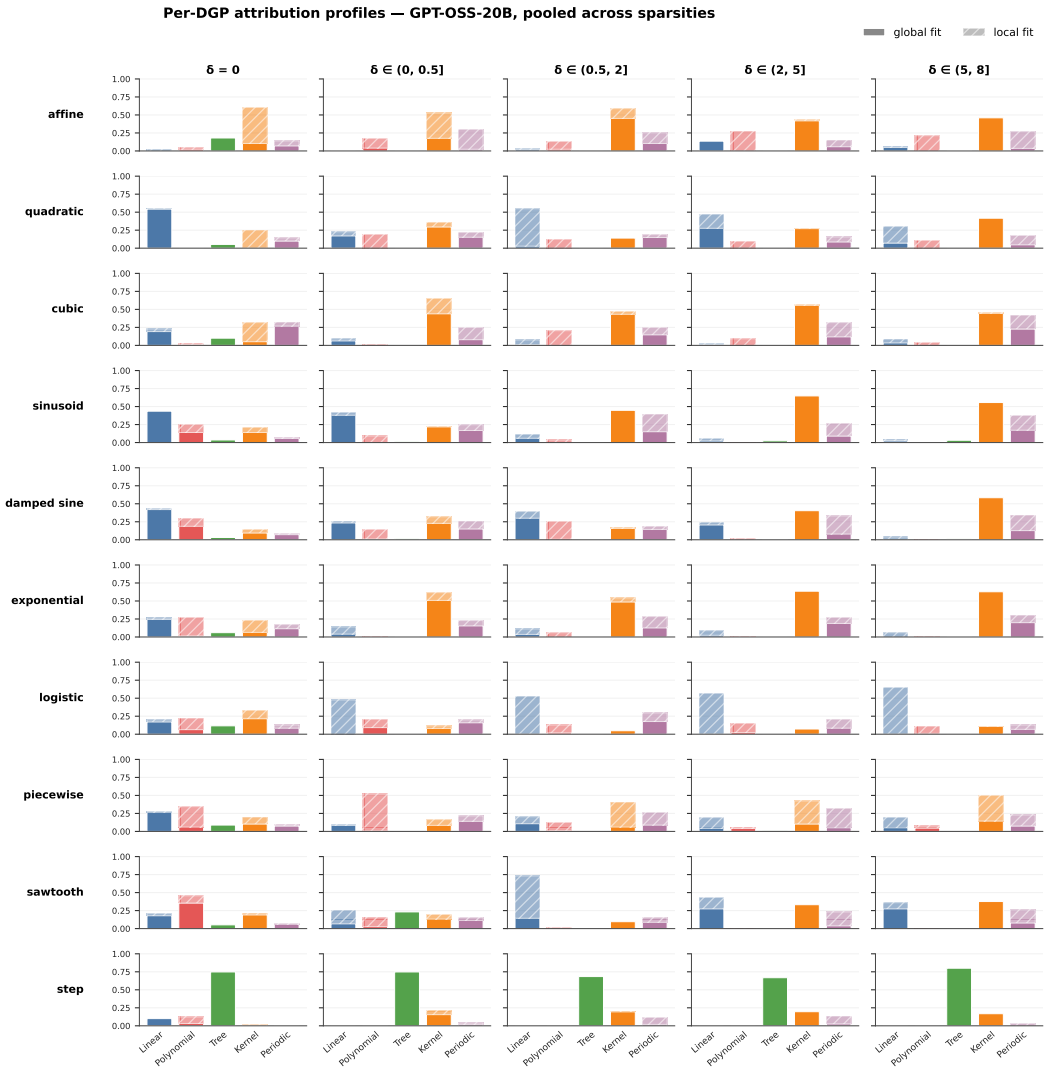

Figure 21: Attribution profiles for *GPT-OSS-20B* across all $\mathcal{G}$ and δ-bins, pooled across sparsities. Solid bar segment: global-fit mass. Hatched bar segment: local-fit mass within the same analytic shape.

δ	Ambiguous queries	$\hat{\rho}_{\text{hard}}$	$\hat{\rho}_{\text{tie}}$	$\hat{\rho}_{\text{filtered}}$
0	64.7%	0.378	0.300	0.414
$(0, 0.5]$	61.8%	0.287	0.287	0.415
$(0.5, 2]$	64.3%	0.344	0.283	0.453
$(2, 5]$	62.7%	0.270	0.292	0.402
$(5, 8]$	60.9%	0.322	0.288	0.383

Table 11: Attribution under hard assignment, shared credit for near-ties, and margin filtering. Shares are computed after pooling queries across models, target functions, and demonstration counts; the filtered profile is normalized over retained queries only. The 0.08 bound concerns these pooled shares, not individual (G, δ) panels.

We use two complementary attribution rules. Margin filtering retains only queries with $M(x^q) > \tau_{\text{cell}}$ and renormalizes the hard attribution profile over these queries. Tie-aware assignment retains all queries and divides credit equally among the families close to the winner:

$$\mathcal{T}(x^q) = \{f : \Delta_f(x^q) \le \Delta_{(1)}(x^q) + \tau_{\text{cell}}\}, \qquad w_f(x^q) = \frac{\mathbf{1}\{f \in \mathcal{T}(x^q)\}}{|\mathcal{T}(x^q)|}.$$

Averaging w_f over queries gives the tie-aware profile. For each rule, $\hat{\rho}$ denotes the largest family share in the resulting profile.

This diagnostic uses the stored divergences for 29 nonconstant predictors grouped into Linear, Polynomial, Local, Tree, Kernel, and Periodic families, with Local retained as a separate family. Table 11 pools the results over the eight models, ten G, and five demonstration counts. Across all five query regimes, 61–65% of queries are ambiguous. Sharing credit changes the pooled dominant-family share by at most 0.08, while restricting to separated queries increases it from 0.27–0.38 to 0.38–0.45. Separation also varies with G: from interpolation to very-far extrapolation, the retained fraction increases from 33% to 62% for cubic and from 43% to 73% for exponential, with median M/τ_{cell} increasing from 0.32 to 1.60 and from 0.70 to 3.37, respectively.

We also calibrate the dominant-family share using predictions from the best single predictor in each cell plus Gaussian noise of scale τ_{cell}, with ten repetitions through the same attribution pipeline. These controls reproduce the pooled shares to within approximately 0.05, with differences up to 0.18 in individual (G, δ) panels. Thus, the magnitude of $\hat{\rho}$ alone does not establish multiple predictive modes. Attribution describes which classical behaviors resemble the LLM predictions, with the margin and tie-aware profile indicating how clearly those behaviors can be distinguished. The family labels remain relative to the chosen library and assignment rule.

H Compute resources

All experiments were run on an internal academic compute cluster with SLURM scheduling. The hardware used was NVIDIA A100 GPUs (40 GB SXM4) on nodes with 256 CPU cores and ≈ 1 TB of system RAM, running CUDA $12.x$ under PyTorch 2.6; storage was a shared filesystem with a global sweep footprint of ≈ 1 GB. No cloud GPUs were used.

The main inference sweep covered 8 models $\times$ 10 $\mathcal{G}$ $\times$ 5 n_{demos} $\times$ 5 δ-bins $= 2000$ cells of 50 test queries each. Per-cell wall-clock at $n_{\text{demos}} = 100$ ranged from 5–20 min for non-reasoning models on 2$\times$ A100-40GB up to ≈ 80 min for GPT-OSS, which uses the longer 16384-token decoding budget required by its chain-of-thought trace. The full sweep consumed approximately 1,100 A100-hours. Pointwise attribution against the 34-algorithm library and the 30-configuration RANSAC peel are CPU-only and complete in roughly 4 hours and 2 hours respectively on a single login node. The steering pipeline of runs on a single A100-40GB, taking between 30 min (GPT-OSS-20B, two-mode case) and 1 h (QWEN-2.5-7B, three-mode case) per (model, $\mathcal{G}$) cell, for an aggregate of $\approx$ 25 GPU-hours across the cells we report.

The reported experiments consume approximately 1,150 A100-hours. Including preliminary runs, debugging iterations (notably a re-run triggered by an initial `max_new_tokens` setting that truncated reasoning-model outputs), and exploratory analyses on layers, peeling thresholds, and attribution

Table 12: Fraction of local windows satisfying Eq. 1, pooled across models, G, and demonstration counts. Noise includes Gaussian and unit-variance Student-t_3 variants; single curve includes smooth-fit and saturating controls. For these two columns, we show the larger variant rate. The planted-branch control uses branch probability 0.3. Fifteen windows flagged on noise-free G because of query-grid aliasing are excluded throughout.

$\delta(x_q)$	Windows	LLM	G	G+noise	Single curve	Planted branches
0	1,405	4.9%	0.0%	0.1%	1.2%	90.2%
$(0, 0.5]$	1,953	15.5%	0.0%	0.4%	0.5%	96.1%
$(0.5, 2]$	2,765	16.7%	0.0%	0.2%	0.9%	95.5%
$(2, 5]$	3,027	18.6%	0.0%	0.1%	1.0%	96.4%
$(5, 8]$	1,961	19.4%	0.0%	0.2%	0.8%	97.1%

libraries that did not make it into the paper, the full project consumed roughly 2,000 A100-hours together with $\sim$ 200 CPU-hours of analysis and plotting.

I LOCAL BRANCH STRUCTURE

The global RANSAC peel can split a single curved function into several pieces because its fitting class is deliberately narrow. We therefore make a local comparison that asks whether nearby predictions in $F_{\mathrm{ICL}}(\cdot \mid D)$ are better described by contiguous pieces of one function or by branches that alternate across queries. For each fixed D, we sort $(x_{qj}, \hat{y}_j)$ by query input and take windows of $W = 30$ points at stride five, without crossing the boundary of the demonstration range. The initial $W = 20$ was increased to 30 after calibration on synthetic surfaces, before examining LLM flag rates. We use the same interpolation and extrapolation bins defined by $\delta(x_q)$ in Section 4. Within a window w, the tolerance is $\tau_w = \max\{0.03 \ \mathrm{range}(\hat{y}_w),\ 0.01 \ \mathrm{range}(y^D)\}$.

Both descriptions use affine functions. The interval description I_2 places one change point in query order and fits a line by least squares on each side, with at least five points per piece. The branch description B_2 finds two lines by sequential RANSAC, then reassigns points to the nearer line and refits twice. Let f_{I_2} and f_{B_2} be the fractions of predictions within τ_w of their respective fitted lines. Let R_2 count runs of the two branch labels in query order after omitting unassigned points, and let h_1, h_2 denote the fitted branches. We flag a window only if

$$f_{B_2} \geq 0.80, \qquad f_{B_2} - f_{I_2} \geq 0.25, \qquad R_2 \geq 4, \qquad |h_1(x_w) - h_2(x_w)| \geq 3\tau_w, \qquad \min_k n_k \geq 5, \tag{1}$$

where x_w is the midpoint of the window's query range and n_k is the number of points assigned to branch k. Thus a flag requires two separated, well-supported continuations that interleave locally; two successive segments of one curve do not suffice. The true function G is not used to fit or flag LLM windows.

We apply the identical rule on the same query grids to G itself, to G with Gaussian or heavy-tailed noise of scale $a + b\delta$ fitted to the LLM's residual scatter, and to two single-curve controls: a smooth fit to the LLM predictions and a saturating continuation of G. We also plant two interleaved affine branches as a positive control. Table 12 shows that flagged LLM windows increase from 4.9% in interpolation to 19.4% in very-far extrapolation, while the corresponding single-function controls remain below 1.3% in every pooled bin (at most 3.6% in any (G, δ) cell). Across the 11,111 eligible windows, 16.0% are flagged; 97 of 328 evaluable $(\mathrm{model}, G, \delta)$ cells exceed the largest matched single-function control rate by at least 0.20. Requiring at least ten windows per cell gives 94 of 309 cells. Moreover, 163 of 305 evaluable $F_{\mathrm{ICL}}(\cdot \mid D)$ surfaces (53%) contain at least two non-overlapping flagged extrapolation windows.

The result is a geometric test under the stated affine locality and tolerance. About half of flagged extrapolation windows contain two nearly constant levels, and 29% of all eligible windows are adequately fit by neither I_2 nor B_2. The planted-branch control misses its preset 90% acceptance level in three in-support cells, where its branches cross or one branch has fewer than five points; conditional on both requirements of the rule being possible, it flags 98.7% of the fixed-probability control windows. This analysis uses strict numeric completions: DeepSeek-R1 yields no evaluable windows, and Llama-3.1-8B and GPT-OSS-20B have fewer than the other models. The local test

therefore establishes branch structure in the evaluable greedy prediction surfaces, without assigning probability weights to the branches.

J COEXISTENCE AND REWEIGHTING OF MODES IN LLM'S OWN OUTPUT

We examine whether several predictive modes receive probability under one fixed demonstration set D, and whether their weights change with the query x^q. This complements the branch structure of $F_{\mathrm{ICL}}(\cdot \mid D)$ by measuring the model's numerical-answer probabilities directly.

Setup and candidate hypotheses. We use Qwen2.5-7B-Instruct, Gemma-3-12B-it, and Llama-3.1-8B-Instruct on all ten target functions G, with five demonstration sets of $n = 100$ examples. For each D, we use 25 discovery queries to identify recurring constants and lines in the numerical-answer distributions. Candidates must be supported at three or more distinct discovery queries. Together with $h_{\mathrm{true}}(x) = G(x)$ and $h_{\mathrm{mean}}(x) = n^{-1}\sum_i y_i^D$, these form a family $\mathcal{H}_D = \{h_k\}$ frozen before evaluation on 50 separate test queries. The test grid contains ten queries in each of the five distance regimes, using the common demonstration support $[-10, 10]$ to set the grid. Demonstrations and their order remain fixed across queries.

For each candidate, define

$$B_k(x^q) = [h_k(x^q) - s_k(x^q), h_k(x^q) + s_k(x^q)], \qquad s_k(x^q) = 0.05 \max\Big\{\max_i y_i^D - \min_i y_i^D,\ |h_k(x^q)|\Big\}.$$

Intersecting bands are merged for probability accounting; their individual candidates are treated as unidentified at that query.

Probability accounting. We expand answer strings in decreasing prefix probability using the model's unmodified token probabilities. Completed responses are parsed with the numerical extractor used in the main experiments, and probabilities of strings yielding the same answer, rounded to two decimal places, are added together. Enumeration uses token and path cutoffs of 10^{-3} and 10^{-4}, a budget of $4{,}000$ expanded prefixes, and a maximum of 24 generated tokens. Pruned paths, unexpanded prefixes, and responses that do not terminate within the token budget retain their probability as unresolved mass.

At query x^q, let a_q denote retained numerical mass, b_q retained non-numerical mass, and ρ_q unresolved mass, so that $a_q + b_q + \rho_q = 1$. Let e_{kq} be the retained mass in $B_k(x^q)$, and let $\mathcal{N}$ denote a numerical answer. The conditional band probability satisfies

$$P\big(B_k(x^q) \mid x^q, D, \mathcal{N}\big) \in [L_{kq}, U_{kq}] = \left[\frac{e_{kq}}{a_q + \rho_q}, \frac{e_{kq} + \rho_q}{a_q + \rho_q}\right].$$

Thus, unresolved probability widens the bounds rather than being assigned to a candidate or discarded by renormalization.

Coexistence and reweighting. We test a mixture over $\mathcal{H}_D$, assuming that each component places at least $1 - \epsilon$ of its numerical-answer probability in its own band, with $\epsilon = 0.05$. We allow a representation-error tolerance $\eta_q = o_q + b_q$, where o_q is retained numerical mass outside every candidate band; both terms are fractions of total output probability. Assuming each band probability differs from its mixture approximation by at most η_q, the admissible weight of an identifiable candidate is bounded by

$$\ell_{kq} = \max\left\{0, \frac{L_{kq} - \epsilon - \eta_q}{1 - \epsilon}\right\}, \qquad u_{kq} = \min\left\{1, \frac{U_{kq} + \eta_q}{1 - \epsilon}\right\}.$$

We compare the 30 test queries in the moderate, far, and very-far extrapolation regimes, excluding queries with $o_q > 0.20$. Coexistence is certified when two identifiable candidates have $\ell_{kq} > 0.05$ at the same query. A fixed-weight mixture is excluded when, for some candidate,

$$\max_q \ell_{kq} > \min_q u_{kq},$$

where the extrema use queries at which that candidate is identifiable. Dominant switching records a change in the candidate whose lower weight bound exceeds the upper bounds of all other identifiable candidates at a query. These comparisons hold D fixed and are conditional on the declared family, concentration assumption, and error tolerance.

Model	D with reweighting	Pairs excluding fixed weights	Mean ρ_q
Qwen2.5-7B-Instruct	$39/50$	$2{,}090/3{,}340$ (63%)	0.086
Gemma-3-12B-it	$27/50$	$702/2{,}019$ (35%)	0.150
Llama-3.1-8B-Instruct	$0/50$	$0/684$	0.568

Table 13: Probability-based comparisons at $n = 100$. Demonstration-set counts pool ten targets and five sets per target. Pair counts use eligible extrapolation queries within the same D whose greedy answers fall in different candidate bands. Unresolved mass is averaged over all 50 test queries. Llama's broad bounds limit identification, so its absence of reweighting certificates does not establish fixed weights.

Results. Table 13 summarizes the fixed-D comparisons. Reweighting is certified for $39/50$ demonstration sets in Qwen and $27/50$ in Gemma. Among tested query pairs assigned to different bands by greedy decoding, 63% and 35%, respectively, exclude fixed weights. Certified coexistence occurs at 8.9% and 2.2% of eligible extrapolation queries, respectively, averaging the within-D rates over demonstration sets and target functions.

At the model–G level, we require reweighting and dominant switching each in at least three of five demonstration sets, after excluding combinations whose median fraction of queries with $o_q > 0.20$ exceeds one half. Three combinations satisfy this criterion: Qwen on sinusoidal and piecewise-linear targets, and Gemma on the exponential target.

Sensitivity. We repeat the analysis at $n = 20$ with three demonstration sets, applying the same fractional thresholds. Across the resulting 60 model–G–n combinations, band scales of 2%, 5%, and 10% yield 2, 6, and 17 classifications with reweighting and dominant switching. Varying ϵ between 0 and 0.10 or doubling η_q changes at most one overall classification and preserves the six positive classifications at the default band scale. Changing the discovery grid alters the discovered family in the tested Qwen affine case. The experiment therefore provides evidence of coexistence and query-dependent probability support within the specified candidate family and resolution.

K Role of training distribution and support

We separate extrapolation beyond the demonstrations from extrapolation beyond the model's training range. By training transformers on known function distributions, we can examine how both the available function classes and the training support affect $F_{\mathrm{ICL}}(\cdot \mid D)$.

Training. Following Garg et al. (2022), we train decoder-only GPT-2-like transformers with 12 layers, 8 attention heads, and hidden dimension 256. Scalar inputs and labels are interleaved through a shared linear embedding, with a linear regression readout at each input position. The context contains 64 input–label pairs, and training minimizes squared error over all prefix predictions. Inputs are sampled uniformly from $[-2, 2]$, with independent label noise $\mathcal{N}(0, 0.1^2)$. We compare two training distributions:

$$\begin{aligned} &\text{T-lin:} && G(x) = wx + b, && w, b \sim \mathcal{U}[-3, 3], \\ &\text{T-mix:} && \text{equal mixture of T-lin and } G(x) = a\sin(\omega x + \phi), \end{aligned}$$

where $a \sim \mathcal{U}[0.8, 2]$, $\omega \sim \mathcal{U}[1, 2.5]$, and $\phi \sim \mathcal{U}[0, 2\pi)$. Each regime uses three independent training seeds. We train with Adam, batch size 64, and learning rate 10^{-4} for 200,000 steps, followed by 100,000 steps with cosine decay to 10^{-5}. On held-out tasks with 63 demonstrations, all six models attain mean squared error within 1.08–1.29 times the numerical posterior-mean reference for their corresponding function classes, below the acceptance threshold of 1.5.

Evaluation. Demonstrations are sampled from $S = [-2, -0.5]$, a strict subset of the training range. We use two target functions that share a value and slope at $x = -1.25$:

$$G_{\sin}(x) = 1.5\sin\big(1.5(x + 1.25)\big), \qquad G_{\mathrm{lin}}(x) = 2.25(x + 1.25).$$

Their similar behavior on S makes sparse, noisy demonstrations ambiguous between the two continuations. For each model, target G, and $n \in \{4, 8, 32\}$, we draw 100 demonstration sets with

Training regime	n	$x^q \in S$	$x^q \in (-0.5, 2]$	$x^q \in (2, 8]$
T-lin	4	$0/72$	$0/72$	$0/72$
T-lin	8	$0/72$	$0/72$	$0/72$
T-lin	32	$0/72$	$0/72$	$0/72$
T-mix	4	$0/72$	$0/72$	$0/72$
T-mix	8	$0/72$	$4/72$	$0/72$
T-mix	32	$0/72$	$0/72$	$0/72$

Table 14: Rejections of unimodality after multiple-testing correction and the secondary-mass requirement. Each denominator comprises twelve queries, two targets, and three training seeds. Tests use 100 demonstration redraws within each trained model; predictions from different training seeds are not pooled.

the same label-noise level as training. Each D is held fixed across 36 queries: twelve in S, twelve in $(-0.5, 2]$, and twelve in $(2, 8]$. These distinguish interpolation, extrapolation within the training range, and extrapolation beyond the training range. Here, normalized distance is measured from the fixed sampling interval, $\delta(x^q) = \mathrm{dist}(x^q, S)/1.5$, placing the training boundary at $\delta = 5/3$.

At each query, we apply the conditional-density test to the 100 predictions across demonstration redraws, separately for each trained model. We use the Silverman critical-bandwidth test with 200 variance-corrected smoothed-bootstrap replicates, after standardization by a pooled kernel-ridge mean and input-dependent variance estimate. Benjamini–Hochberg correction is applied at 0.05 across the 36 queries within each (model, G, n) combination, and a secondary mode must contain at least 5% of predictions. The continuous regression readout requires no dequantization. All variation at a fixed query comes from the demonstrations; predictions are deterministic given (x^q, D).

Results. Within the training range, T-lin generally continues a fitted linear relation beyond S. T-mix produces a broader range of intermediate predictions between the two continuations when demonstrations are ambiguous, with this spread contracting as n increases. Table 14 summarizes the modality tests. T-lin has no rejections among 648 query-level tests. T-mix has four, all for G_{lin} at $n = 8$ in one training seed, at adjacent queries $x^q \in [1.375, 2]$ ($\delta \in [1.25, 5/3]$). The secondary groups contain 43–45% of predictions. No rejections occur at $n = 4$ or $n = 32$.

The largest qualitative changes occur beyond the training boundary $x = 2$, where predictions depart from the class continuations and approach bounded outputs. Different training seeds reach different output levels, but neither regime has a unimodality rejection in this region when tested per model. Thus, a change at the training boundary need not imply switching between multiple modes within one model. These controlled results show that leaving the demonstration support alone is insufficient for widespread switching, and that both the training function distribution and its input support shape extrapolation behavior.

L Conditional-density test for multiple modes marginalized over demonstrations

This test looks for multiple groups of predictions without fitting branches to $F_{\mathrm{ICL}}(\cdot \mid D)$. For each of Qwen2.5-7B, Llama3.1-8B, and Gemma3-12B, and each of the ten target functions G, we fix 50 query locations x^q (ten in each δ-bin). At each location we draw 100 independent demonstration sets $D^{(r)}$ of size $n = 100$ and collect the greedy predictions $\hat{y}^{(r)} = F_{\mathrm{ICL}}(x^q \mid D^{(r)})$. The model, G, query, prompt template, and decoding are fixed within a location. Thus the 100 predictions sample the output distribution induced by varying D, with no decoding randomness. The full test comprises $3 \times 10 \times 50 \times 100 = 150{,}000$ prompts.

For each (model, G) cell, we fit an RBF kernel-ridge mean $\hat{\mu}(x^q)$ and a smooth residual scale $\hat{\sigma}(x^q)$ to the pooled predictions. At each fixed query we form $z^{(r)} = (\hat{y}^{(r)} - \hat{\mu}(x^q))/\hat{\sigma}(x^q)$ and test whether its distribution has at most one mode using Silverman's critical-bandwidth test with 200 smoothed-bootstrap replicates. We add uniform jitter in $[-0.005, 0.005]$ at the output scale before testing to resolve ties from two-decimal answers. A location counts as having multiple prediction

G	Qwen2.5-7B	Llama3.1-8B	Gemma3-12B	Mean
Affine	0.58	0.54	0.66	0.59
Quadratic	0.58	0.64	0.64	0.62
Cubic	0.52	0.28	0.60	0.47
Exponential	0.40	0.62	0.68	0.57
Sinusoid	0.56	0.54	0.46	0.52
Damped sine	0.60	0.84	0.58	0.67
Sawtooth	0.70	0.54	0.74	0.66
Piecewise linear	0.64	0.30	0.56	0.50
Logistic	0.22	0.46	0.24	0.31
Step	0.26	0.74	0.24	0.41
Mean	0.51	0.55	0.54	0.53

Table 15: Fraction of the 50 query locations with a detected secondary prediction group for each (model, G) cell. Each location uses 100 fresh demonstration sets; rejections are controlled within each cell at Benjamini–Hochberg $\alpha = 0.05$ and require secondary-group mass of at least 5%.

groups only when the test rejects after Benjamini–Hochberg correction over the 50 locations in its cell at $\alpha = 0.05$, and the largest secondary density group contains at least 5% of predictions. This procedure specifies no functional form for the groups and does not use the RANSAC peel.

Table 15 gives the fraction of query locations meeting both criteria. Across the 30 model-G cells, 798 of $1{,}500$ locations (53%) have more than one detected prediction group. The rate varies by target: it averages 67% for damped sine and 66% for sawtooth, compared with 31% for logistic and 41% for step. For affine targets it rises from 43% within the demonstration range to 70% in the nearest extrapolation bin. Across all targets, the corresponding in-support and pooled extrapolation rates are 51% and 54%. These results supply evidence of predictive multiplicity across demonstration draws. Because a new D is used for each repetition, this test alone does not show that several modes coexist under one fixed D or that the query changes their probability weights.

M LIMITATIONS

We study one-dimensional scalar regression with noise-free demonstrations because this setting provides a controllable and visually verifiable prediction surface on which query-conditioned mode switching can be resolved with high precision, closely paralleling the function-learning paradigm in cognitive science. However, this leaves open whether the same phenomena generalize to broader forms of in-context learning such as classification, language understanding, multi-step reasoning, or higher-dimensional regression. Likewise, our attribution framework relies on a finite and non-exhaustive vocabulary of classical inductive biases. Pointwise attribution assigns each query to the closest member of this vocabulary, but behaviours that fall outside the chosen families are necessarily approximated by the nearest in-vocabulary mode. We therefore interpret attribution behaviourally rather than mechanistically: the framework characterises what predictions resemble, not the algorithms internally implemented by the model. Although steering experiments provide causal evidence that distinct predictive modes correspond to separable directions in the residual stream, the framework does not explain where these modes originate.

Several additional limitations arise from the peeling and evaluation procedures themselves. The number and shape of recovered modes depend on analyst choices such as the hypothesis class, residual tolerance, and decomposition order, meaning that different configurations can yield different mode partitions. We mitigate this by applying the same decomposition procedure symmetrically to the synthetic baselines and by focusing on deviations relative to those baselines, but detailed claims about exact mode counts remain configuration-dependent. Developing a principled generative account linking demonstrations, similarity structure, and pretraining statistics to emergent mode structure remains an important direction for future work.