How Context Shapes Internal Representations in Tabular Foundation Models

Anonymous authors
Paper under double-blind review

Abstract

Tabular foundation models make predictions by learning statistical structure from examples provided in context. However, little is known about how this structure is represented internally or how it varies with the amount of examples provided. We study this using controlled synthetic inference problems. These problems contain exact reference posteriors whose geometry varies along known properties, such as covariance orientation, correlation, skewness, and mode separation. Across TabPFN V2, V2.5, V3, and TabICL, these properties are recoverable from low-dimensional hidden-state subspaces. We discover that intervening along the same subspaces causally changes model predictions. Copying only the task-subspace component of a query state from one context into another reproduces 95–99% of the resulting change in the final prediction, while the remaining component reproduces almost none. Small encoded shifts produce substantial movement in the intended direction, which is increased when the model is provided with longer context. Larger requested shifts are only partially realized at intermediate depths. We find that more context makes the controlled settings easier to decode from hidden states. Finally, training short-context models to match long-context task coordinates successfully moves their representations toward the long-context targets, but does not consistently improve prediction beyond training on long-context outputs alone. Output-only training can even reduce prediction error while moving the task coordinates away from those targets. We conclude that representations can be both decodable and causally steerable without providing additional predictive value as training targets.

1 Introduction

The tabular foundation model ecosystem has expanded rapidly in the past few years. Open-weight models, including several generations of TabPFN (Hollmann et al., 2023; 2025; Grinsztajn et al., 2025; 2026) and TabICL (Qu et al., 2025), perform on par with or better than many established methods without task-specific training. These models make predictions through in-context learning. In this paradigm, one supplies labeled examples as context, and the model predicts for new queries in a single forward pass. While the predictive performance of these models has been widely studied, less is known about how they internally represent the statistical structure they infer from context, or how those representations shape their predictive distributions.

Research on transformers suggests that task-relevant information can take structured and causally useful forms in hidden states. In language models, task and function vectors extracted from demonstrations can induce task-specific behavior when inserted into a separate forward pass (Hendel et al., 2023; Todd et al., 2024), while transformers trained on controlled stochastic processes encode the geometry of Bayesian belief states in their hidden activations (Shai et al., 2024). In tabular foundation models, probes recover regression coefficients and intermediate quantities from hidden states (Gupta et al., 2026b), and causal interventions reveal distinct prediction mechanisms across architectures (Biloš et al., 2026). Most closely related, contrastive steering directions estimated on one data split with TabPFN-2.5 did not reduce prediction error on held-out samples (Gupta et al., 2026a).

These results leave several distinct questions about what it means for a model to represent a task. First, information about a statistical property may be decodable from hidden states without playing a causal role in prediction. Second, even when intervening on a representation changes the corre-

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.

sponding decoded property, it may change other aspects of the predictive distribution as well. Third, a representation that is decodable and causally effective need not provide a useful target for learning. These distinctions are particularly relevant for in-context learning because the representation is formed from a finite set of examples. We therefore ask whether identifiable representations of statistical properties can be recovered across independently sampled contexts, whether interventions on those representations causally alter predictions, and whether those same representations provide useful supervision when learning from less context.

To study these questions, we build on the use of tabular foundation models as conditional density estimators for simulation-based inference (Vetter et al., 2025). We construct controlled synthetic inference problems with exact reference posteriors whose distributional properties vary along known coordinates. These include the orientation of a covariance matrix, the strength of a correlation, the skewness of a posterior, and the separation of its modes. Selected posterior summaries, such as the mean, are held fixed while the designated property varies. Because the reference posterior is known exactly, we can directly compare the model's predictive distribution with it. We also draw independent contexts and vary context length using nested prefixes of the same draw, which changes the amount of evidence available while holding the underlying problem fixed. Across TabPFN V2, V2.5, V3, and TabICL, we examine whether these distributional properties can be decoded from hidden states in held-out contexts and settings, how predictions respond when we intervene on the corresponding representations (Figure 1), and whether representations formed with more context provide useful training targets for models given less.

We first find structured, low-dimensional representations of these distributional properties in all four models. Representations calibrated on one set of contexts remain decodable in held-out contexts. With more context, states from the same setting sit closer together relative to states from different settings, for every model and task. We then intervene along the same independently calibrated subspaces. Small interventions generally move predictions in the requested direction, and the response becomes stronger with more context. For example, a requested 5° Gaussian rotation moves the output by 3.12° for TabPFN V3 and 3.40° for TabICL with 512 context rows, increasing to 3.72° and 4.17° with 2,048 rows. Steering is also strongly depth-dependent. For a 67.5° request, TabPFN V3 moves by 30.4° at approximately three-quarters depth but by 67.1° at the final block. Replacing the intermediate hidden state with the model's native state from the target setting produces a similar result, suggesting that incomplete steering at intermediate depths is not simply a consequence of imperfectly estimating the task subspace. A cross-context patching experiment explains this pattern. Copying only the task-subspace component of a query state from a source context into a base context recovers nearly as much of the source prediction as copying the full state, and 95–99% of it at the final block. At intermediate depths, however, even the full source state recovers only part of it, and for bimodality and skew almost none until the last blocks. The task information is therefore concentrated in the identified subspace, but is not settled in the query row until late in the network. For sufficiently large interventions, the decoded coordinate can also move away from the requested value even while the full predictive distribution moves closer to the model's native prediction at the target setting. Taken together, these results suggest that the identified coordinates have a causal effect on predictions but that control through any single decoded coordinate is incomplete.

Finally, we ask whether these representations provide useful supervision rather than only a means of interpreting or intervening on a pretrained model. We construct matched short- and long-context examples in which a frozen teacher receives 2,048 rows and a student receives the first 512 rows (and is therefore nested in the long-context), with both models answering the same query. We compare training the student to reproduce the teacher's predictive distribution with training that additionally aligns the student's hidden state with the teacher within the identified task subspace. The additional representation objective reliably moves the student's task coordinates toward those produced from the longer context, reducing that distance in 31 of 32 model–task–depth conditions on held-out examples. This does not translate into better prediction. Guided and output-only training reduce error to the exact posterior by similar amounts, and for TabICL guidance does slightly worse in every condition. The main exception is TabPFN V3 on the Gaussian task, where output-only training increases joint error at every depth and guidance avoids this only for early-block updates. Output-only training also often improves prediction while moving the task coordinates further from the long-context targets. For TabICL on the Gaussian task with all-block updates, error falls by 15% while task-coordinate distance rises by 126%. The identified coordinates are therefore effective

targets for changing internal states, but matching them is neither necessary for nor reliably additive to the predictive gains from long-context supervision.

2 Related Work

Our work sits at the intersection of prior-fitted networks, model interpretability, representation-guided adaptation, and in-context learning. Prior-fitted networks motivate viewing tabular foundation models as performing approximate posterior inference in context (Müller et al., 2022; Nagler, 2023), including their use as conditional density estimators for simulation-based inference (Vetter et al., 2025). Work on task and function vectors, probing, and activation steering motivates our use of hidden-state decoding and intervention (Hendel et al., 2023; Todd et al., 2024; Hewitt & Manning, 2019; Belinkov, 2022; Turner et al., 2023; Rimsky et al., 2024). Recent work has begun applying related interpretability methods directly to TabPFN (Gupta et al., 2026b;a; Biloš et al., 2026). We build on these strands of the literature by studying known statistical properties of exact reference posteriors, calibrating their representations across independent contexts, and testing both their causal effect on predictions and their value as supervision for short-context adaptation. Our final experiment further connects to knowledge and context distillation (Hinton et al., 2015; Romero et al., 2015; Snell et al., 2022) by asking whether matching an identified task representation provides information beyond matching a long-context teacher's outputs. We discuss these connections in greater detail in Appendix A.

3 Experimental Setup

We study how TabPFN V2, V2.5, V3, and TabICL represent distributional information using four synthetic task families with known reference posteriors. Each family has a scalar control parameter τ that governs a property of the distribution. A context $D = \{(x_i, \theta_i)\}_{i=1}^{n}$ contains n simulated examples, with independent draws indexed by seed ε. Given this context and query inputs q, the model predicts a distribution over the target θ. We compare its prediction $\hat{p}$ with the reference posterior $p^*_{\tau,q}$, and extract a coordinate Y summarizing the predicted property, such as the rotation angle implied by its covariance.

Appendix table 3 summarizes the task families. Appendix B.1 gives the simulators and reference distributions; Table 4 collects calibration and evaluation settings.

Gaussian rotation keeps queries fixed as the context-generating setting changes. Scalar contexts contain examples spanning parameter values supplied as input features; we hold each context fixed and vary the query, including its parameter value. During a steering intervention, the context and query inputs remain fixed.

A separate cross-context patching experiment on TabPFN V3 and TabICL varies the task setting through the context while holding queries fixed for all four families. Its scalar generators omit the control parameter from the input features and use different joint distributions that preserve the specified conditional families. Labels and output decoding share a fixed target scale across contexts, with model weights frozen. Appendix B.7 describes this design.

We distinguish changes in the decoded coordinate Y from changes in the predictive distribution. Coordinate movement measures the response in the selected property. Distributional comparisons measure either accuracy against the reference posterior or agreement with the model's native prediction at the target setting; each experiment specifies which comparison it uses.

Let C_ℓ and H_ℓ denote the hidden states of the context rows and query rows after block ℓ. We report depth as $\ell/(L-1)$ for a model with L blocks. For TabPFN V3 and TabICL, H_ℓ is the full query-row state. TabPFN V2 and V2.5 use a separate token for each feature and for the target, so H_ℓ contains feature tokens unless stated otherwise.[1] In all four architectures, context rows attend only to other context rows. An intervention on H_ℓ, thus, leaves C_ℓ and later context states unchanged. Downstream query layers, however, continue to attend to them. Figure 1 summarizes this structure.

[1]This distinction is necessitated by the architectural differences between second and third generation TabPFN models.

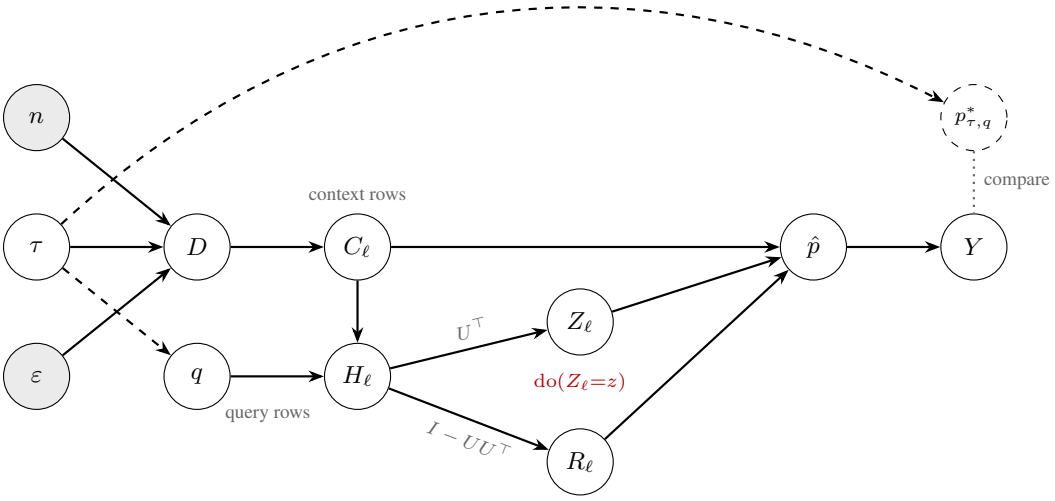

Figure 1: Causal structure of the intervention experiments. The task setting τ enters through context D or through query q. At block ℓ, we intervene on the query-row state H_ℓ, which is split into task coordinates $Z_\ell = U^\top H_\ell$, with U calibrated on separate contexts (Section 4), and a residual R_ℓ. The decoded output Y is compared with the analytic posterior $p^*_{\tau,q}$. Gray nodes are fixed by design.

4 Decoding

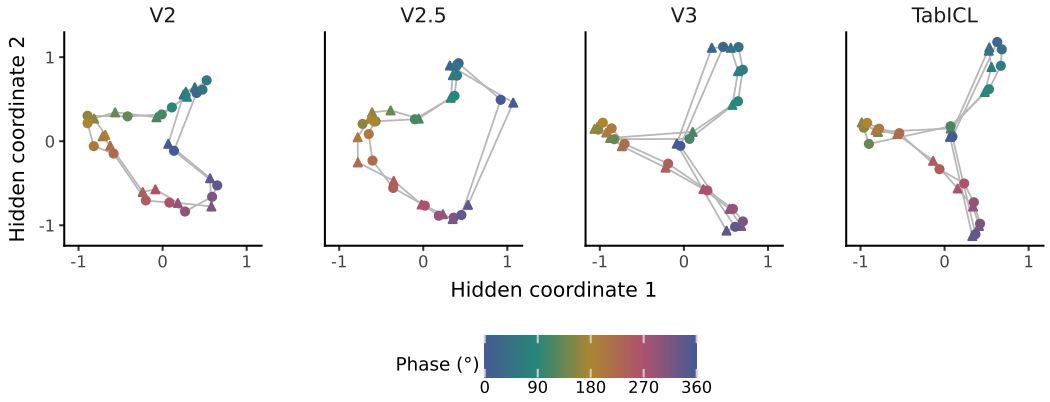

Figure 2: **Circular hidden representations.** Hidden-state trajectories of TabPFN V2, V2.5, V3, and TabICL across 16 phases of the 3D Gaussian rotation (circles and figure-eights).

We identify task representations from states from calibration contexts, which are disjoint from evaluation contexts. For each task we fix A settings $\tau_1, \ldots, \tau_A$, equally spaced over the task's range. Let $h_{s,a} \in \mathbb{R}^d$ be the query-row state H_ℓ for calibration context $s \in \{1, \ldots, S\}$ at setting τ_a. We center each context across settings and average across contexts,

$$m_a = \frac{1}{S} \sum_{s=1}^{S} \Big(h_{s,a} - \frac{1}{A} \sum_{a'=1}^{A} h_{s,a'} \Big), \qquad a = 1, \ldots, A, \tag{1}$$

to remove context-specific offsets and keep variation driven by τ. We then stack m_a as rows of $M \in \mathbb{R}^{A \times d}$ and take the singular value decomposition $M = W \Sigma V^\top$. The task basis $U \in \mathbb{R}^{d \times k}$

consists of the first k right singular vectors, and task coordinates are $Z_\ell = U^\top H_\ell$. The anchor coordinates $c(\tau_a) = U^\top m_a$ define a calibration curve, interpolated with a cubic spline to obtain $c(\tau)$. Table 4 and Appendix B.2 specify the calibration grids, ranks, and evaluation samples for each experiment. The two-dimensional projections in Figure 2 are fitted separately for visualization; interventions use the corresponding k-dimensional basis.

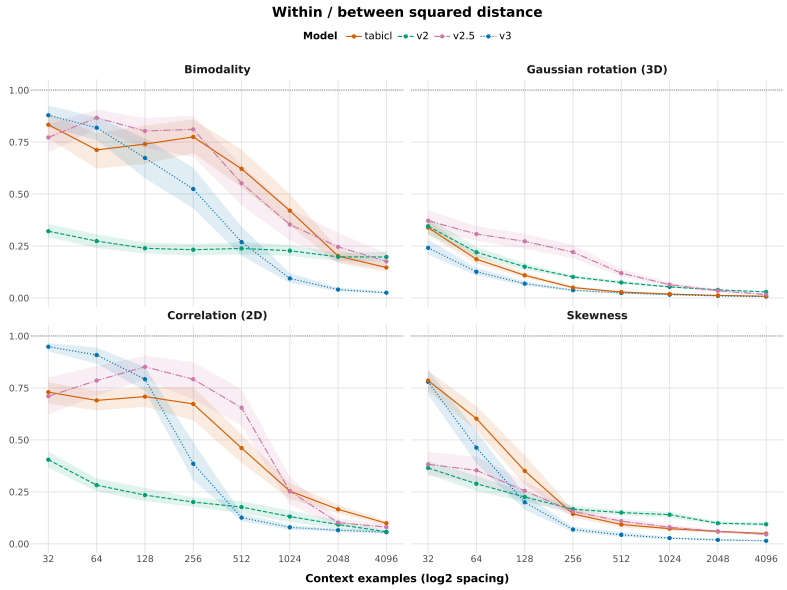

Figure 3: **More context brings same-setting states together.** Ratio of the mean squared task-coordinate distance between held-out states at the same τ to that between states at different τ, from 32 to 4,096 context rows; the dotted line marks a ratio of one and bands are pointwise 95% context-bootstrap intervals.

Results. We identify 16 fixed task settings by assigning each held-out state to its nearest independently calibrated centroid in task coordinates (Appendix B.4). Chance accuracy is 6.25%, and shuffling the τ labels gives at most 19%. With 2,048 context rows, accuracy ranges from 44% to 100% across all four models and tasks (Figure 5). The number of rows required depends on the task. TabPFN V3 and TabICL, for example, identify the rotation phase almost perfectly from 512 rows. However, for bimodality at the same context length, the four models identify the mode separation in only 17–48% of the evaluation contexts. More context also brings states with the same τ closer together compared to states with different τ. For every model and task, the ratio of these two squared distances is lower at 4,096 rows than at 32 rows (Figure 3).

5 Steering

Decoding demonstrates that task information is accessible in hidden states. To test whether changing it affects predictions, we intervene within the independently calibrated task subspace while holding the context and query inputs fixed. For a held-out context at setting b, we request a move to $t = b+\delta$ by replacing the query-row state after block ℓ with

$$H'_\ell = H_\ell + U\big(c(t) - c(b)\big), \tag{2}$$

where U and $c(\tau)$ are the basis and calibration curve from Section 4. This changes the task coordinates while preserving the residual $(I - UU^\top)H_\ell$. Later query layers continue to attend to the unchanged context.

We measure movement relative to the identity intervention as $\Delta Y = \mathrm{sign}(\delta)(Y_{\mathrm{steer}} - Y_{\mathrm{id}})$, using the shortest signed angular difference for Gaussian rotations. Positive values indicate movement in the requested direction. We also measure changes in absolute target-coordinate error, since directional

movement need not improve accuracy. Controls include identity patches, which must reproduce native predictions, norm-matched random directions, piecewise-linear interpolation between anchors, and native target-state replacement.

Small steps follow the request, more so with more context. For a 5° Gaussian rotation request, V3 realizes 3.12° with 512 rows and 3.72° with 2,048; TabICL realizes 3.40° and 4.17°. Norm-matched random directions produce near-zero mean movement. Scalar tasks respond less uniformly across models (Table 5; Appendix B.5).

Large steps stall at intermediate depth. At 2,048 rows, a 67.5° request moves V3's output by 30.4° at depth 0.75 but by 67.1° at the final block. Replacing the query state with the model's own state at the target setting yields a similar 30.6°, so the shortfall is not an artifact of the estimated subspace. For TabPFN V2 and V2.5, our default patches only feature tokens, which understates their response relative to patching all query tokens (Appendix B.6.2).

The task subspace carries the context's effect, but only late in the network. A second experiment patches states between contexts rather than adding a steering vector, an interchange intervention in the sense of Geiger et al. (2024). We run the model on a source context and a base context generated at different settings of τ, with the same queries, and copy part of the source query state into the base run after block ℓ (Appendix B.7). At the final block, copying only the task-subspace component closes 94.8–98.9% of the squared-Hellinger gap between the two predictions in model–task pairs passing the native checks, exceeding the residual component and matched controls. At every depth, the task component tracks the full source state closely. At intermediate depths, however, even the full source state recovers only part of the gap, and for bimodality and skew almost none until the last blocks (Appendix Figure 19). This explains the depth dependence of steering: the task information is concentrated in the identified subspace but is not settled in the query row until late in the network.

The decoded coordinate is an incomplete summary. At a 135° request at depth 0.75 with 2,048 rows, the decoded angle for V3 and TabICL moves against the request on average, while their predictions move closer in squared Hellinger distance to the model's own prediction at the requested setting (Appendix Figure 13). A single decoded coordinate can therefore misstate an intervention's effect on the full predictive distribution.

These interventions establish causal sensitivity to the identified coordinates. Section 6 asks whether matching those coordinates also improves short-context predictions beyond training on long-context outputs alone.

6 Learning From Long-Context Representations

Longer contexts provide the models with more evidence about the data-generating process and often produce more consistent task representations (figures 3 and 9). We therefore ask whether the identified representations can themselves provide supervision for short-context adaptation. Specifically, can a model given a short context be adapted to reproduce the predictive behavior and internal task representation that the unmodified model forms from a matched long context? Appendix figure 20 summarizes this long-to-short training setup. Relative to Figure 1, the task coordinates are no longer modified directly during inference. Instead, fine-tuning changes the mapping from the short context to the student's query-row state H_ℓ^S, while the prediction and task coordinates produced from the matched long context serve as training targets. Previous work has attempted to use hidden state vectors or internal representations during the model training process Kangaslahti et al. (2026). To our knowledge this has not previously been applied to tabular foundation models. This experiment therefore tests a stronger claim than decodability or steering alone, namely whether the identified task coordinates provide a useful target for learning.

For each simulated dataset, we construct a 512-row short context and a matched 2,048-row long context, with the short context nested within the long context. A frozen copy of the pretrained model processes the long context and serves as the teacher. A student receives the short context and is adapted using low-rank adaptation (LoRA) (Hu et al., 2022). Crucially, both models receive the same query, but they differ in their context. The teacher provides targets for both the predictive

Table 1: Update depth and distance to long context. Entries are mean percentage changes in squared task-space distance to the paired native-2048 state, relative to each fit's native-512 baseline.

			Change in distance to long context (%)			
Model	Objective	Updated blocks	Gaussian 3D	Bimodality	Skew	Correlation
TabPFN V3	Output-only	6 (0.25)	+451.3	−15.2	+59.3	+4.7
		12 (0.50)	+416.4	+60.7	+59.0	+4.2
		17 (0.75)	+307.7	+7.1	+20.7	−1.1
		All ICL blocks	+41.5	−22.1	+22.4	+1.4
	Guided	6 (0.25)	−14.8	−45.9	−21.5	+0.4
		12 (0.50)	−16.8	−47.6	−20.7	−0.02
		17 (0.75)	−11.9	−24.6	−9.9	−2.5
		All ICL blocks	−19.2	−52.9	−31.5	−3.8
TabICL	Output-only	3 (0.25)	−2.5	−8.2	+0.7	−41.1
		6 (0.50)	+20.7	−7.5	+14.2	−43.1
		8 (0.75)	+105.2	+6.0	+21.8	−36.4
		All ICL blocks	+125.6	−4.9	+20.6	−55.7
	Guided	3 (0.25)	−44.1	−11.8	−18.2	−51.1
		6 (0.50)	−45.2	−9.3	−18.4	−51.5
		8 (0.75)	−42.7	−10.5	−6.4	−43.3
		All ICL blocks	−48.0	−23.5	−19.9	−64.8

distribution and the query-row hidden state. The analytic posterior is reserved for evaluation and is not used to optimize or select the student.

Specifically, Let p_T and h_T denote the teacher prediction and query-row hidden state, and let p_S and h_S denote the corresponding student quantities. Given an independently calibrated task basis U, guided adaptation minimizes output distillation together with task-coordinate alignment,

$$\mathcal{L}_{guided} = \mathrm{KL}(p_T \| p_S) + \lambda_z \frac{\|U^\top (h_S - h_T)\|_2^2}{s_z} + \lambda_r \frac{\|(I - UU^\top)(h_S - h_S^0)\|_2^2}{s_r},$$

where h_S^0 is the native short-context state and s_z and s_r normalize the corresponding initial training-set gaps. The first term encourages the student's posterior predictive distribution to match the teacher's. The second encourages the student's hidden state to match the teacher's within the identified task subspace. The third penalizes changes outside that subspace, keeping the student close to its native short-context representation. We compare guided adaptation with output-only distillation and with several representation controls, including rank-matched random directions, shuffled representation targets, and dense hidden-state alignment. We restrict adaptation to rank-8 LoRA updates at selected transformer blocks. LoRA holds the pretrained weights fixed and learns low-rank updates to selected weight matrices, substantially reducing the number of trainable parameters relative to full fine-tuning (Hu et al., 2022). This restriction makes matched sweeps across objectives, tasks, and seeds computationally feasible.

Table 2: Predictive error before and after guided adaptation in the four-task scope experiment. The adapters are the late-block and all-block guided adapters of Table 1. Entries are means [minimum, maximum] over three fitted seeds. TV is total-variation error to the exact posterior (%); Δ TV is adapted minus native 512 (percentage points). N–A TV is predictive mass moved by adaptation (percentage points).

Model	Task	Updated blocks	Native TV (%)	Adapted TV (%)	Δ TV (pp)	N–A TV (pp)
TabPFN V3	Gaussian 3D	17	4.86 [4.51, 5.11]	5.12 [4.80, 5.29]	0.26 [0.18, 0.31]	3.36 [3.24, 3.59]
TabPFN V3	Gaussian 3D	All ICL	4.86 [4.51, 5.11]	4.79 [4.36, 5.10]	-0.07 [-0.20, 0.14]	3.94 [3.62, 4.20]
TabPFN V3	Bimodality	17	7.88 [7.57, 8.21]	6.12 [5.96, 6.45]	-1.75 [-1.89, -1.61]	5.07 [4.75, 5.24]
TabPFN V3	Bimodality	All ICL	7.88 [7.57, 8.21]	5.84 [5.51, 6.12]	-2.03 [-2.09, -1.95]	4.99 [4.58, 5.50]
TabPFN V3	Skew	17	6.20 [6.08, 6.37]	5.26 [5.00, 5.49]	-0.94 [-1.15, -0.80]	3.10 [2.61, 3.43]
TabPFN V3	Skew	All ICL	6.20 [6.08, 6.37]	5.21 [4.78, 5.57]	-0.99 [-1.37, -0.79]	3.09 [2.96, 3.32]
TabPFN V3	Correlation	17	7.12 [6.89, 7.37]	6.65 [6.58, 6.69]	-0.47 [-0.69, -0.20]	3.17 [2.87, 3.48]
TabPFN V3	Correlation	All ICL	7.12 [6.89, 7.37]	6.23 [6.20, 6.24]	-0.90 [-1.13, -0.65]	3.17 [2.99, 3.36]
TabICL	Gaussian 3D	8	4.60 [4.51, 4.75]	4.33 [4.10, 4.49]	-0.27 [-0.40, -0.14]	1.21 [0.99, 1.36]
TabICL	Gaussian 3D	All ICL	4.60 [4.51, 4.75]	4.05 [3.95, 4.11]	-0.55 [-0.65, -0.44]	1.80 [1.77, 1.87]
TabICL	Bimodality	8	7.13 [6.83, 7.67]	6.94 [6.60, 7.46]	-0.19 [-0.28, -0.07]	2.69 [2.11, 3.69]
TabICL	Bimodality	All ICL	7.13 [6.83, 7.67]	6.02 [5.60, 6.50]	-1.10 [-1.23, -0.91]	3.61 [3.36, 3.86]
TabICL	Skew	8	6.43 [6.17, 6.57]	5.32 [5.20, 5.44]	-1.11 [-1.24, -0.98]	2.80 [2.57, 3.02]
TabICL	Skew	All ICL	6.43 [6.17, 6.57]	5.20 [5.09, 5.34]	-1.23 [-1.40, -1.09]	3.17 [2.92, 3.61]
TabICL	Correlation	8	13.73 [13.30, 14.39]	9.58 [8.99, 10.24]	-4.14 [-4.87, -3.06]	10.85 [10.19, 11.32]
TabICL	Correlation	All ICL	13.73 [13.30, 14.39]	8.04 [7.58, 8.46]	-5.69 [-6.33, -4.84]	11.39 [11.18, 11.79]

We find that guided fine-tuning generally moves short-context representations closer to their matched long-context targets. In the six-seed Gaussian comparison in Figure 21, mean squared task-coordinate distance decreases by 13.3% for TabPFN V3 and 39.3% for TabICL. Table 1 reports the separate three-seed scope/depth experiment comparing output-only and guided adaptation; its settings are collected in Table 4. Within the late single-block and all-block conditions, guided adaptation reduces task-coordinate distance in all 16 model–task–scope conditions, with larger reductions from all-block updates in all eight matched model–task comparisons. All-block LoRA also has a larger parameter budget, so this comparison combines the effects of update placement and adaptation capacity.

Table 2 reports the separate three-seed primary analysis of one prespecified guided condition for each model–task pair. Guided adaptation reduces total-variation error to the exact posterior in all but one setting (the single block fine-tuning of TapPFN V3 on the Gaussian rotation task), with reductions ranging from -0.26 to 5.69 percentage points. The largest reductions occur for TabICL, but both models generally show a modest improvement of approximate 1%.

The direct change in the predictive distribution is larger than the resulting reduction in posterior error. For example, TabPFN V3's all block correlation adaptation moves 3.17 percentage points of predictive mass while reducing error by only 0.90 points. In comparison, TabICL's all block correlation adaptation moves 11.39 points while reducing error by 5.69 points. Thus, only part of the predictive change moves the model closer to the exact posterior.

These improvements relative to the native short-context model suggest that the benefits from representation guidance are modest at best. The guided objective also contains the same output-matching term used by output-only adaptation. In the matched factorial experiments, representation guidance improves long-context state matching by construction, but does not consistently outperform output-only adaptation on predictive metrics. Broader updates across the ICL stack improve predictive performance more consistently than adding the representation objective. The evidence therefore supports the use of task coordinates as effective targets for changing internal states, but does not establish a general predictive advantage over output-only adaptation.

7 Discussion and Limitations

Across four models, the distributional properties we vary can be decoded from low-dimensional subspaces of the query-row hidden states, and shifting those coordinates steers predictions. These representations provide a causal pathway through which we can intervene in the models, but the relationship between these representations and model behavior in the wild remains correlational. The control they provide is, however, local in three senses. It is local in size: small requested changes are realized, while large ones are reached only in part at intermediate depths (Figure 13).

It is local to the coordinate: the decoded property can move away from a request while the full predictive distribution moves toward the target, so a single coordinate is an incomplete summary of an intervention. And it is local to the context: which examples a model sees shapes both its representations and their consequences.

Guidance moved the student's task coordinates toward the teacher's in 31 of 32 conditions (Table 1) but did not consistently improve prediction over output-only distillation (Table 6). One possible reason is that part of what the long-context teacher knows comes from rows the short-context student never sees, so its coordinates partly encode information the student cannot learn from the short context alone. A second is that the coordinates, although they carry nearly all of the context's effect at the final block of the frozen model, are not necessary once its weights change: output-only training often reduces error while moving the coordinates further from the teacher's, so adaptation can reach teacher-like predictions through a different internal route. Averaging targets over many long contexts that share a short prefix would test the first explanation, and recalibrating the task subspace on the adapted model would test the second.

Relation to in-context learning. Although our setting is specific to tabular foundation models and posterior properties, it bears on task representations in in-context learners more broadly. As with task and function vectors in language models (Hendel et al., 2023; Todd et al., 2024), the context's effect is summarized in a low-dimensional subspace of the query state, and exact posteriors let us check what that subspace encodes. One difference matters for interpreting such interventions: task vectors are typically patched into runs without informative demonstrations, whereas our interventions compete with a context that remains present and that later layers continue to read, which may explain why steering is local and why the context's effect settles late. That matching these representations adds nothing beyond output distillation suggests they are more useful for interpreting and controlling in-context learning than as extra supervision for distilling it (Snell et al., 2022).

Limitations. Our conclusions are limited to four pretrained models, three TabPFN generations and TabICL, and four low-dimensional synthetic task families with known posteriors. We do not test real data or high-dimensional tasks. It is also possible that the low dimensional representations we describe are sufficient for steering interventions, but unused during inference, as previous work has noted Makelov et al. (2024). Most interventions compare contexts of 512 and 2,048 rows. A sweep from 32 to 4,096 rows shows that task representations become less variable (figures 3 and 9) with more context for TabPFN V3 and TabICL. However, the correlation we identify is not monotonic, particularly for TabPFN V2 or V2.5. Thus, we do not claim that longer context stabilizes representations in general. We intervene only on query rows and for TabPFN V2 and V2.5 mainly on feature tokens. An ablation demonstrates that this default may understate the steerability of TabPFN V2 and V2.5. Mediation conclusions require the base model to reproduce the reference posterior, which TabICL narrowly fails for bimodality at 512 rows. Fine-tuning uses rank-eight LoRA on TabPFN V3 and TabICL only, without an extensive hyperparameter sweep, so the adaptation results need not be optimal; the all-block condition also has more trainable parameters, confounding update placement with capacity. Conditional and joint predictive metrics can disagree. Finally, our constructed intervals are pointwise bootstrap intervals over held-out contexts without correction for multiple comparisons.

8 Conclusion

In this work, we used synthetic problems with exact posteriors to interrogate how tabular foundation models represent tasks. In all models, task properties can be decoded from low-dimensional representations of the hidden states. Adjusting these subspaces steers predictions, and copying them between contexts carries nearly all of the context's effect on the final prediction. This control nevertheless weakens for large changes, and further depends on which examples are in context. A single decoded coordinate may also misstate an intervention's effect on the full distribution. Training a short-context model to match long-context representations adjusted them as intended, but predicted no better than training only on long-context outputs. One explanation is that part of what the long-context model knows comes from rows the short-context model never sees. Further work could test this by distilling toward targets averaged over long contexts that share a short prefix, and ask whether these findings extend to other in-context learners.

AI USE STATEMENT

We used AI in the form of LLMs to assist in the creation of this project. Most notably, most of the code used to carry out the experiments was implemented using coding agents. More specifically, both Claude Code (Anthropic) and Codex (OpenAI) were used to implement the experiments. The code was implemented and refined over the course of 2026, and multiple generations of language models were used. Although implementations were machine assisted, the team developed the experiments presented in this paper based on the literature review process and prior research experience.

The initial draft of the manuscript was written by the research team. LLM-based tools were used to edit and refine the draft. LLM-based tools were used to during later stages of writing to help reorganize the appendix. Some of the writing in the appendix (outside of the related work and simulator details sections) was partially generated by LLM-based tools, which added expanded upon earlier drafts written by the research team. These appendix additions were manually checked.

AI/ML tools were used during the literature review process, but not primarily in the form of chatbots or coding agents. The team used services like ScholarInbox to keep track of reference lists, and search for related work. Language models were used during the literature review process by individual members of the team to help digest papers added to reference list as needed.

REPRODUCIBILITY STATEMENT

We have included a code supplement along with our submission that contains the files required to reproduce the results presented in this manuscript. We provide and `UV` environment and lock file for environment replication.

The original experiments were run using `Slurm`. All experiments were run on NVIDIA GPUs. In order to anonymize the code and avoid assuming specific hardware or system requirements, we do not include the original `Slurm` scripts used by the research team. Instead, we provide scripts that run the code directly using whatever resources are available.

We intend to release all of the data we collected during the research process as soon as possible. We were unable to include it in the code supplement due to the 100MB upload cap.

REFERENCES

Mohammed Abdullah. Understanding context sampling in TabPFN on small tabular datasets, 2026.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76930–76966. Curran Associates, Inc., 2024. doi: 10.52202/079017-2447.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023. URL `https://openreview.net/forum?id=0g0X4H8yN4I`.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Workshop Track Proceedings*, 2017. URL `https://openreview.net/forum?id=HJ4-rAVtl`.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Marin Biloš, James T. Wilson, Anderson Schneider, and Yuriy Nevmyvaka. A mechanistic study of tabular foundation models. *arXiv preprint arXiv:2605.21288*, 2026. URL `https://arxiv.org/abs/2605.21288`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel

Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359.

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. TuneTables: Context optimization for scalable prior-data fitted networks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 83430–83464. Curran Associates, Inc., 2024. doi: 10.52202/079017-2654.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598, 2022.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 160–187. PMLR, 2024.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the state of the art in tabular foundation models, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3: Technical report, 2026.

Atharva Gupta, Dhruv Kumar, Murari Mandal, and Saurabh Deshpande. Where computation lives inside TabPFN: Causal localisation of attention head function, 2026a. Workshop on FMSD at ICML 2026.

Aviral Gupta, Armaan Sethi, and Dhruv Kumar. TabPFN through the looking glass: An interpretability study of TabPFN and its internal representations. *arXiv preprint arXiv:2601.08181*, 2026b. URL https://arxiv.org/abs/2601.08181.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*

the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. NIPS 2014 Deep Learning Workshop.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=a7BFdumEVD.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jiachen Jiang, Yuxin Dong, Jinxin Zhou, and Zhihui Zhu. From compression to expression: A layerwise analysis of in-context learning, 2025.

Sara Kangaslahti, Nihal V. Nayak, Jonathan Geuter, Marco Fumero, Francesco Locatello, and David Alvarez-Melis. Boomerang distillation enables zero-shot model size interpolation. In *International Conference on Learning Representations (ICLR)*, 2026.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530, 2023.

Yuxuan Li, Declan Campbell, Stephanie C. Y. Chan, and Andrew Kyle Lampinen. Just-in-time and distributed task representations in language models, 2025.

Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32287–32307. PMLR, 2024.

Jan-Matthis Lueckmann, Jan Boelts, David S. Greenberg, Pedro J. Gonçalves, and Jakob H. Macke. Benchmarking simulation-based inference. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 343–351. PMLR, 2021.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? An interpretability illusion for subspace activation patching. In *International Conference on Learning Representations (ICLR)*, pp. 26486–26515, 2024.

Calvin McCarter. What exactly has TabPFN learned to do?, 2025. Originally published in the ICLR 2024 Blogposts Track.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022.

Sarthak Mittal, Niels Leif Bracher, Guillaume Lajoie, Priyank Jaini, and Marcus Brubaker. Amortized in-context Bayesian posterior estimation, 2025.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25660–25676. PMLR, 2023.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. ICLR: In-context learning of representations. In *International Conference on Learning Representations (ICLR)*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847. PMLR, 2025.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full Bayesian inference in context? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51531–51582. PMLR, 2025.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Ivan Rubachev, Akim Kotelnikov, Nikolay Kartashev, and Artem Babenko. On finetuning tabular foundation models, 2025.

David Rundel, Julius Kobialka, Constantin von Crailsheim, Matthias Feurer, Thomas Nagler, and David Rügamer. Interpretable machine learning for TabPFN. In *Explainable Artificial Intelligence (xAI 2024)*, volume 2154 of *Communications in Computer and Information Science*, pp. 465–476. Springer, Cham, 2024. doi: 10.1007/978-3-031-63797-1_23.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, volume 37, pp. 75012–75034, 2024. doi: 10.52202/079017-2387.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022.

Aditya Tanna, Pratinav Seth, Mohamed Bouadi, and Vinay Kumar Sankarapu. Exploring fine-tuning for tabular foundation models. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 8613–8616. ACM, 2026. doi: 10.1145/3774904.3792923.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 108439–108467. Curran Associates, Inc., 2024. doi: 10.52202/079017-3442.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023.

Julius Vetter, Manuel Gloeckler, Daniel Gedon, and Jakob H. Macke. Effortless, simulation-efficient Bayesian inference using tabular foundation models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 82685–82724, 2025. doi: 10.52202/085713-2770.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962, 2024. doi: 10.52202/079017-2041.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphael Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior, 2026.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, and Robert D. Nowak. Task vectors in in-context learning: Emergence, formation, and benefits. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=lODGn1Rp5t.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, pp. 87400–87442, 2025.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations (ICLR)*, pp. 1651–1678, 2024.

Tian Zhou, Beverly Jin, Linxiao Yang, Xue Wang, Wenwei Wang, Bingqing Peng, Mengni Ye, Jinjie Gu, and Liang Sun. What do tabular foundation models compute in context? In-situ representation refinement through attention-gated updates, 2026.

A Related Work

A.1 Prior-fitted networks and tabular foundation models.

Prior-data fitted networks (PFNs) are trained on datasets sampled from a prior. Their in-context predictions approximate the Bayesian posterior predictive (Müller et al., 2022). Nagler (2023) provides a complementary frequentist account of their behavior. TabPFN first applied PFNs to small tabular classification problems (Hollmann et al., 2023), and later versions extended the approach to general tabular prediction and scaled data size (Hollmann et al., 2025; Grinsztajn et al., 2025; 2026). TabICL (Qu et al., 2025) also performs tabular prediction with a column-then-row architecture. Transformers, in context, can also perform full Bayesian inference for common statistical models (Reuter et al., 2025). Objectives and architectures for amortized in-context posterior estimation have been compared systematically (Mittal et al., 2025). In simulation-based inference (SBI) (Cranmer et al., 2020; Lueckmann et al., 2021), TabPFN can be used as a simulation-efficient conditional density estimator (Vetter et al., 2025). We build directly on this interpretation of tabular foundation models as in-context distribution estimators. That is, rather than evaluating only predictive accuracy, we construct problems with known posterior families so that we can ask how specific properties of the inferred distribution are represented internally and manipulated by the model.

A.2 Model Interpretability

Existing interpretability work on TabPFN has studied the model at several levels. Feature-attribution methods have been adapted to its in-context setting (Rundel et al., 2024), black-box analyses have characterized the functions it learns (McCarter, 2025), and hidden-state probes recover regression coefficients and intermediate statistical quantities (Gupta et al., 2026b). Biloš et al. (2026) use causal interventions to identify differences in prediction mechanisms across tabular foundation model architectures. We build on this work by focusing on distributional properties with known ground-truth coordinates, which allows us to relate hidden-state geometry directly to a controlled change in the target posterior.

Most closely related, Gupta et al. (2026a) use activation patching and ablations to localize attention-head function in TabPFN-2.5 and report that contrastive steering vectors estimated on one data split do not improve prediction on held-out samples. Our experiments similarly attempt to steer each model but instead by calibrating representations of known posterior coordinates using independent contexts and then intervening on those coordinates in held-out contexts. These interventions do causally move predictions in the requested direction. While seemingly at-odds with their findings, the representations, interventions, evaluation targets differ in our experimental setup. In fact, our adaptation experiments recover a related distinction. Representation guidance changes internal task coordinates as intended, but does not consistently improve prediction beyond matching the teacher's outputs alone. Together, these results separate whether a representation can causally steer model behavior from whether supervising that representation yields additional predictive gains.

More broadly, probes test what information can be recovered from internal representations (Alain & Bengio, 2017; Hewitt & Manning, 2019). However, this does not establish that the decoded information is used by the model (Hewitt & Liang, 2019; Elazar et al., 2021; Belinkov, 2022). To address this, causal mediation and activation-patching methods can be used to intervene on internal computation to test its behavioral relevance (Vig et al., 2020; Meng et al., 2022; Zhang & Nanda, 2024; Geiger et al., 2024), although subspace interventions can themselves create behavior through mechanisms not used in the native computation (Makelov et al., 2024). Activation steering modifies behavior by adding directions identified from contrasting activations (Turner et al., 2023; Rimsky et al., 2024; Li et al., 2023). When the represented quantity is inherently multidimensional or periodic, a single linear direction may be insufficient (Engels et al., 2025), motivating methods that instead follow structured activation manifolds (Wurgaft et al., 2026). These literatures motivate our separation of decoding from intervention. We first identify low-dimensional task subspaces using calibration contexts and then test their causal relevance by intervening within the same subspaces on held-out contexts. Unlike approaches that steer along a single direction, we use multidimensional subspaces that can capture more complex task geometry, including the periodic structure induced by covariance rotation.

A.3 Representation Guided Fine-Tuning

Tabular foundation models can be adapted by (1) tuning a condensed learned context (Feuer et al., 2024), (2) fine-tuning on retrieved local neighborhoods (Thomas et al., 2024), or (3) full fine-tuning, which Rubachev et al. (2025) find most practical for TabPFNv2. However, the benefits of fine-tuning are not uniform across models and datasets, and it can worsen accuracy or calibration in some settings (Tanna et al., 2026). Parameter-efficient approaches instead modify a pretrained model through low-rank weight updates (Hu et al., 2022) or interventions on hidden representations (Wu et al., 2024).

Our adaptation experiment is more closely related to teacher-student distillation. Standard knowledge distillation trains a student to match a teacher's outputs (Hinton et al., 2015), while representation distillation additionally aligns intermediate hidden states (Romero et al., 2015). Related work has used representation alignment to guide training in other architectures (Yu et al., 2025), while context distillation trains a model with less context to reproduce behavior obtained from a richer context (Snell et al., 2022). We combine these ideas in a long-to-short setting in which a frozen teacher receives a longer context and a LoRA-adapted student receives a matched short prefix. Both, however, receive the same query. Our baseline treatment trains the student to match the teacher's predictive distribution, while representation-guided training additionally aligns the student's hidden state with the teacher within the independently identified task subspace and constrains movement outside that subspace. This design lets us ask specifically whether the task representation provides useful supervision beyond the information already contained in the teacher's output.

A.4 In-Context Learning

In-context learning allows transformers to perform new tasks from demonstrations without parameter updates (Brown et al., 2020), with performance continuing to improve as additional examples are provided in some settings (Agarwal et al., 2024). ICL can be viewed theoretically as implicit Bayesian inference over latent variables (Xie et al., 2022) or as implementing learned estimators such as least squares and gradient descent (Garg et al., 2022; Akyürek et al., 2023; von Oswald et al., 2023). Both views are particularly relevant to our setting because tabular foundation models must infer statistical relationships from the examples we supply in context.

Recent work also suggests that this inferred task information can become explicitly represented in model activations. Induction heads support in-context pattern completion (Olsson et al., 2022), task and function vectors can recover demonstrated behavior (Hendel et al., 2023; Todd et al., 2024), and in-context vectors can partially replace demonstrations through activation steering (Liu et al., 2024). Identified task vectors can also be manipulated to improve robustness (Yang et al., 2025), while transformers trained on hidden Markov processes have been shown to linearly encode Bayesian belief states (Shai et al., 2024). Work has also examined how these internal representations change as additional context is provided (Jiang et al., 2025; Li et al., 2025; Park et al., 2025), including context-size effects in tabular models (Abdullah, 2026; Zhou et al., 2026). We build on this literature by studying task representations when the relevant statistical property is known by construction. This allows us to track how accurately a posterior coordinate can be decoded as context grows and, separately, whether manipulating that coordinate changes the model's predictive distribution.

B Additional Experimental Details

This section specifies the simulators and reference posteriors behind each task family, and the calibration and evaluation protocols used in each experiment.

B.1 Simulation Details

The following simulators define the task families in Table 3 for decoding, context-length, steering, and adaptation experiments. Rows (x_i, θ_i) are drawn independently to form context, and the known generating distributions determine the reference posterior at each query. The separate cross-context patching design is described in Appendix B.7.

Table 3: Task families used in decoding, context-length, steering, and adaptation experiments. Fixed quantities refer to the reference query; Gaussian means are joint posterior means at zero observation. In the separate cross-context patching experiment, all four settings enter through context (Appendix B.7).

Family	Properties varied	Held fixed	Setting enters via
Gaussian rotation	Covariance orientation	Mean, covariance eigenvalues	Context
Correlation	Dependence and scale	Joint mean	Query
Skew	Asymmetry and scale	Mean	Query
Bimodality	Mode separation and modality	Mean, variance, mixture weights	Query

Control parameters. For the 3D Gaussian rotation, a separate context is simulated at each τ. We reuse the same standard normal draws so that contexts differ through τ. For the other three families, τ is drawn independently for each context row and supplied as a feature. In these families an independent standard-normal feature is appended to each row as a nuisance input. It is set to zero in queries.

3D Gaussian rotation. We define the target as $\theta \sim \mathcal{N}(0, I_3)$ and an observation as $x = A_\phi\theta + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_3)$,. Let $A_\phi = DR_\phi^\top$, $D = \mathrm{diag}(2.5, 1.1, 0.4)$, and R_ϕ be the rotation by angle ϕ about the axis $(1,1,1)/\sqrt{3}$. The posterior is $\mathcal{N}(\Sigma_\phi A_\phi^\top x, \Sigma_\phi)$ with $\Sigma_\phi = (I + A_\phi^\top A_\phi)^{-1} = R_\phi\,\mathrm{diag}(0.138, 0.452, 0.862)R_\phi^\top$. The control parameter is the phase $\phi \in [0°, 360°)$. Changing ϕ rotates the posterior covariance while leaving its eigenvalues, the posterior mean at $x = 0$, and the marginal distributions of θ and x unchanged. Since the models predict a scalar target, the joint posterior is queried autoregressively as: θ_1 from x; θ_2 from (x, θ_1); and θ_3 from (x, θ_1, θ_2). Queries set $x = 0$, with $\theta_1 \in \{-1.2, 0, 1.2\}$ for the second fit and $(\theta_1, \theta_2) \in \{-1, 0, 1\}^2$ for the third. The predicted conditional means and variances produce a joint Gaussian. Its covariance is then compared with Σ_ϕ, where the decoded phase is the ϕ whose Σ_ϕ is closest, on a $0.25°$ grid.

Correlation sweep. We define the target as $\theta \sim \mathcal{N}(0, I_2)$ and the observation as $y = \theta + \eta$ with $\eta \sim \mathcal{N}(0, R_\rho)$ and $R_\rho = 0.75^2 \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$. $\rho \sim \mathcal{U}(-0.85, 0.85)$ is drawn per row. The posterior is then $\mathcal{N}\big((I + R_\rho)^{-1}y,\ (I + R_\rho^{-1})^{-1}\big)$. Rows contain features $(y_1, y_2, \rho, \text{nuisance}, \theta_1)$ and target θ_2. The model predicts the conditional posterior of θ_2 given θ_1. Queries set $y = 0$, $\rho = \tau$, and $\theta_1 \in \{-1.5, 0, 1.5\}$ times the posterior standard deviation of θ_1. The posterior mean at $y = 0$ is fixed at zero. As ρ moves from -0.85 to 0.85, the conditional slope of θ_2 on θ_1 moves from -0.735 to 0.735, the conditional standard deviation (0.600 at $\rho = 0$, 0.367 at $|\rho| = 0.85$) and the marginal posterior variance of θ_1 (0.360 to 0.294) also change.

Skew. We define the target is $\theta \sim \mathcal{N}(0, 1)$ and the observation ias $y = g_c(\theta) + \nu$ with $\nu \sim \mathcal{N}(0, 0.35^2)$ and $g_c(\theta) = (e^{c\theta} - 1)/c$ (with $g_0(\theta) = \theta$). The curvature $c \sim \mathcal{U}(-0.9, 0.9)$ is drawn per row. Rows have features $(y, c, \text{nuisance})$. Queries set $c = \tau$ and choose y so that the posterior mean is exactly zero, which requires $|y| \leq 0.153$. The posterior is evaluated on a grid of 3,001 points on $[-5, 5]$. Changing c changes the skewness of the posterior from 0.99 at $c = -0.9$ to -0.99 at $c = 0.9$.

Bimodality. Each row draws a separation $\lambda \sim \mathcal{U}(0, 0.95)$ and a sign $s \in \{-1, 1\}$ with equal probability, then sets $\theta = s\,m_\lambda + \sigma_\lambda z$ with $z \sim \mathcal{N}(0, 1)$, $m_\lambda = 2\lambda/(1 + \lambda^2)$, and $\sigma_\lambda^2 = 2(1 - \lambda^2)/(1 + \lambda^2)$, and observes $y = \theta + \mathcal{N}(0, 2)$. Rows have features $(y, \lambda, \text{nuisance})$. These constants are chosen so that at $y = 0$ the posterior is $\frac{1}{2}\mathcal{N}(-\lambda, 1 - \lambda^2) + \frac{1}{2}\mathcal{N}(\lambda, 1 - \lambda^2)$. Queries set $y = 0$ and $\lambda = \tau$. The posterior mean (0), variance (1), and mixture weights are thus fixed, while the shape changes from a single mode to two. The posterior is bimodal when $\lambda > 1/\sqrt{2}$.

Contexts and outputs. In the steering and fine-tuning experiments, a short context is a prefix of its matched long context. The cross-context patching experiment and geometry figure draw contexts of the stated size directly. For the scalar families, predictions and reference posteriors are compared as probabilities on fixed bins. The decoded coordinate is the setting whose exact posterior is closest

in squared Hellinger distance over a grid of 1,001 settings. Context sizes and calibration grids are listed in Table 4.

B.2 Experimental Protocols

Bases and intervention sites. The context sweep evaluates 32, 64, 128, 256, 512, 1,024, 2,048, and 4,096 rows at requested relative depth 0.75. It uses a fixed rank-7 measurement basis for every family. Steering uses rank seven for Gaussian rotation and rank eight for scalar families. Cross-context patching uses the full numerical rank of its centered calibration anchors, at most seven or eight respectively, with bases refitted at each depth. Its nine requested relative depths are $0, 1/8, \ldots, 1$, and labels and output decoding share target normalization with mean zero and scale one. These are experiment-specific basis choices, not estimates of an intrinsic representation dimension.

Adaptation studies. The six-seed primary comparison covers Gaussian rotation, bimodality, and skew; the three-seed scope/depth comparison additionally includes correlation. Evaluation uses held-out contexts and unseen task settings. Both studies retain frozen task bases read out at V3 block 17 and TabICL block 11, with rank seven for V3 Gaussian rotation and rank eight otherwise. The scope/depth study updates one block (V3: 6, 12, or 17; TabICL: 3, 6, or 8) or all ICL blocks (24 for V3; 12 for TabICL). Single-block parameter counts are matched within each model; all-block LoRA has more parameters. All use rank-8 LoRA, which is distinct from task-basis rank. The late-block/all-block study comprises 96 fits; the quarter- and half-depth extension adds 96. Repeats on full B200 GPUs use the same seeds and do not increase the number of independent fitted seeds.

Uncertainty. Native decoding, steering, and transfer intervals resample whole evaluation contexts, keeping task settings and intervention pairs together. Adaptation intervals resample fitted seeds after averaging their nested contexts and settings. Reported bootstrap intervals are pointwise 95% intervals, conditional on the checkpoint and calibration, without multiplicity correction. The three-seed tables report descriptive means; these do not establish equivalence between objectives.

B.3 Additional Figures

The figure below illustrates, on the Gaussian rotation task alone, an effect summarized in the main text: longer contexts produce less variable task representations.

B.4 Context Sweeps

We evaluate all four models at eight context lengths from 32 to 4,096 rows, reading task coordinates at depth 0.75 in a fixed basis calibrated on separate 4,096-row contexts; the first four figures show representations and predictions becoming more consistent and more accurate as context grows. The last four figures repeat native steering and evaluate the guided adapters of Section 6 across the same range, using only two held-out contexts per cell, and are preliminary.

Table 4: Protocols for the quantitative analyses. Counts are per model–task condition. Calibration contexts are disjoint from evaluation contexts; adaptation contexts are nested within fitted seeds. The context sweep and steering cover all four models, while cross-context patching and adaptation use TabPFN V3 and TabICL.

Experiment	Context rows	Calibration / training	Evaluation and uncertainty
Context sweep	32–4,096	8 contexts at 4,096 rows; 16 settings; frozen rank-7 basis	32 held-out contexts; paired context bootstrap
Small-step steering	512; 2,048	4 contexts; 16 Gaussian phases or 17 scalar settings	12 held-out contexts; paired context bootstrap
Gaussian distance / depth steering	512; 2,048	4 contexts; 16 phases; separate basis for each depth and token scope	12 held-out contexts; paired context bootstrap
Cross-context patching	512	8 contexts; 8 Gaussian phases or 9 scalar settings	12 held-out contexts; 4 transfer pairs at 9 depths; context bootstrap
Primary guided adaptation	512 student; 2,048 teacher	Frozen task basis; 16 training and 4 validation contexts per fit	6 fitted seeds; 8 test contexts per fit; fitted-seed bootstrap
Adaptation scope / depth	512 student; 2,048 teacher	Frozen task basis; 16 training and 4 validation contexts per fit	3 paired fitted seeds; 8 test contexts per fit; descriptive seed summaries
Post-adaptation tails / steering	512; native 2,048 reference	Saved scope adapters; independently calibrated native-2,048 steering curve held fixed	3 fitted seeds; 16 fresh contexts per seed; descriptive seed means

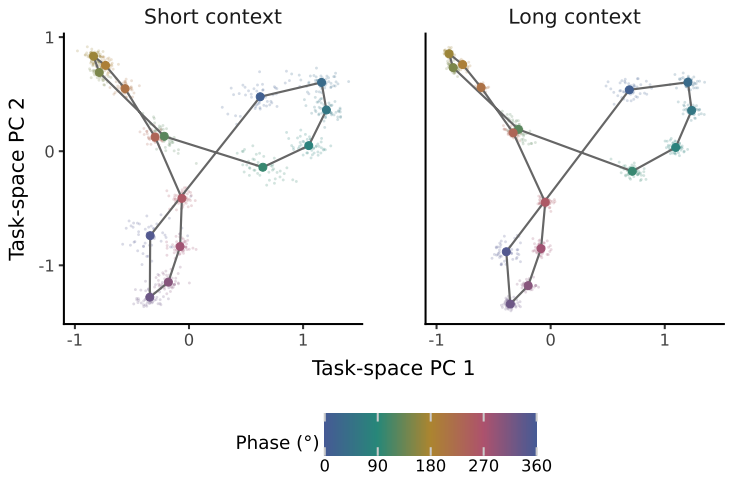

Figure 4: **Longer contexts tighten task representations.** TabPFN V3 query states at block 17 for 16 held-out Gaussian rotation phases, projected from the task subspace onto two shared principal components (small points are individual contexts; large points are phase means); across all seven task coordinates, within-phase variance falls from 10.21 with 512 rows to 5.06 with 2,048.

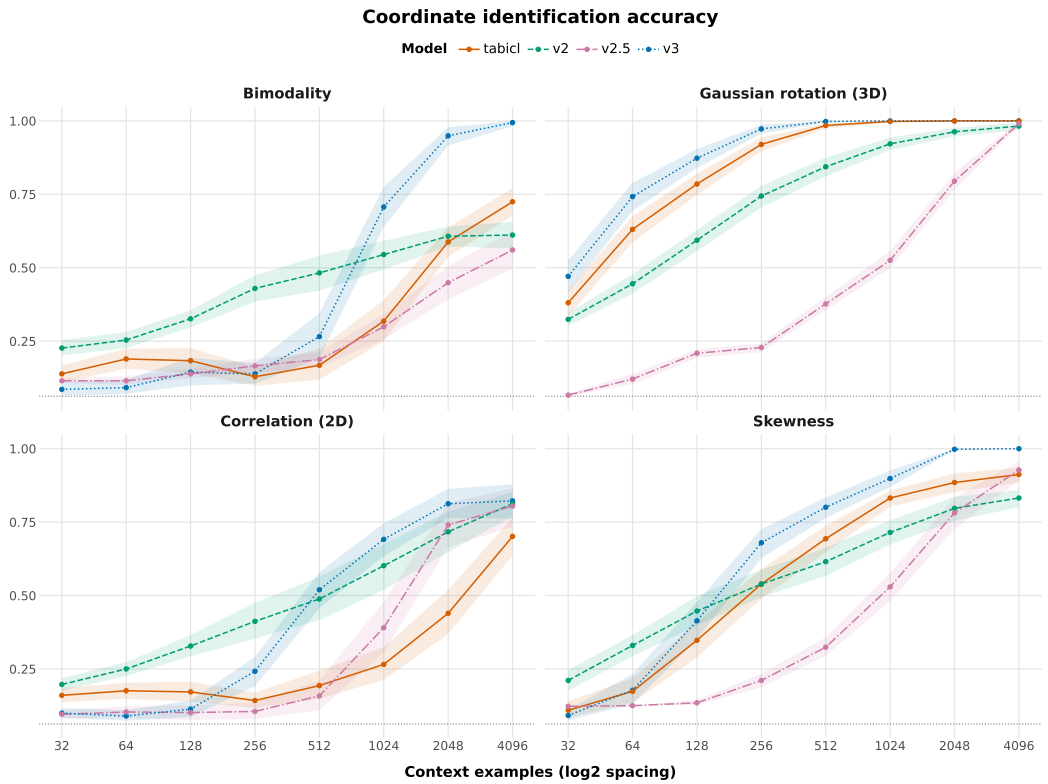

Figure 5: **Task settings become identifiable with more context.** Fraction of held-out states assigned to the correct one of 16 settings by their nearest independently calibrated centroid in task coordinates; the dotted line marks chance (6.25%) and bands are pointwise 95% context-bootstrap intervals.

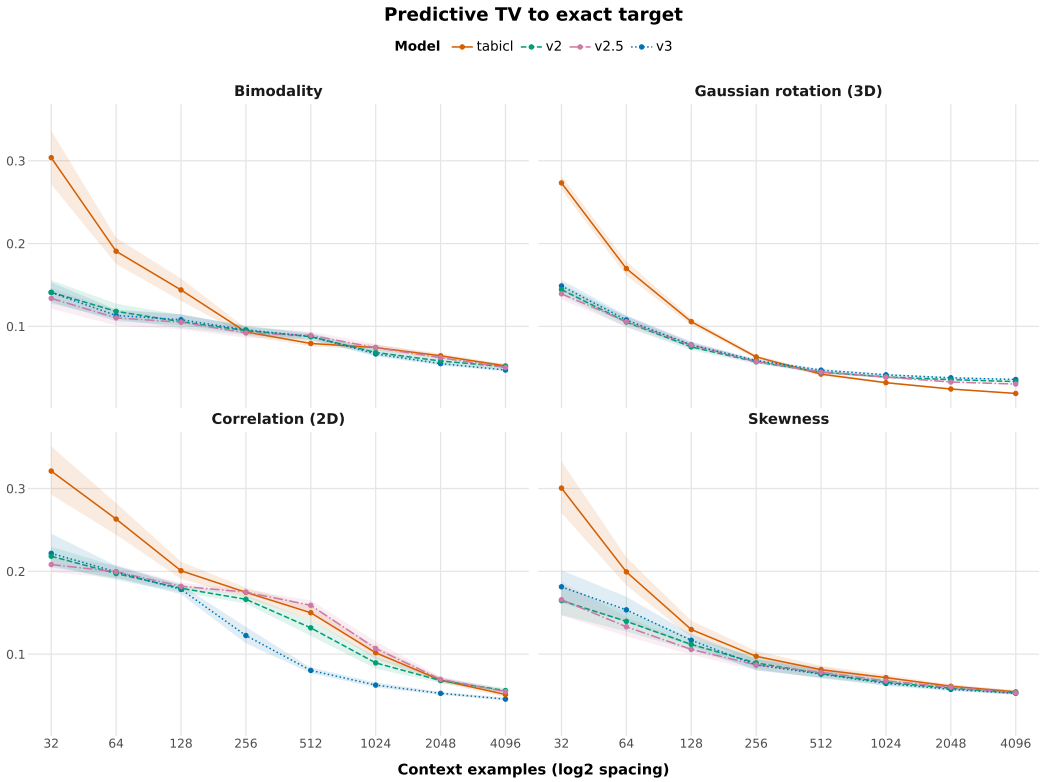

Figure 6: **Predictions approach the exact posterior with more context.** Total-variation distance from each model's predictive distribution to the exact reference posterior; TabICL is least accurate with short contexts but is comparable to the TabPFN models from roughly 256 rows onward.

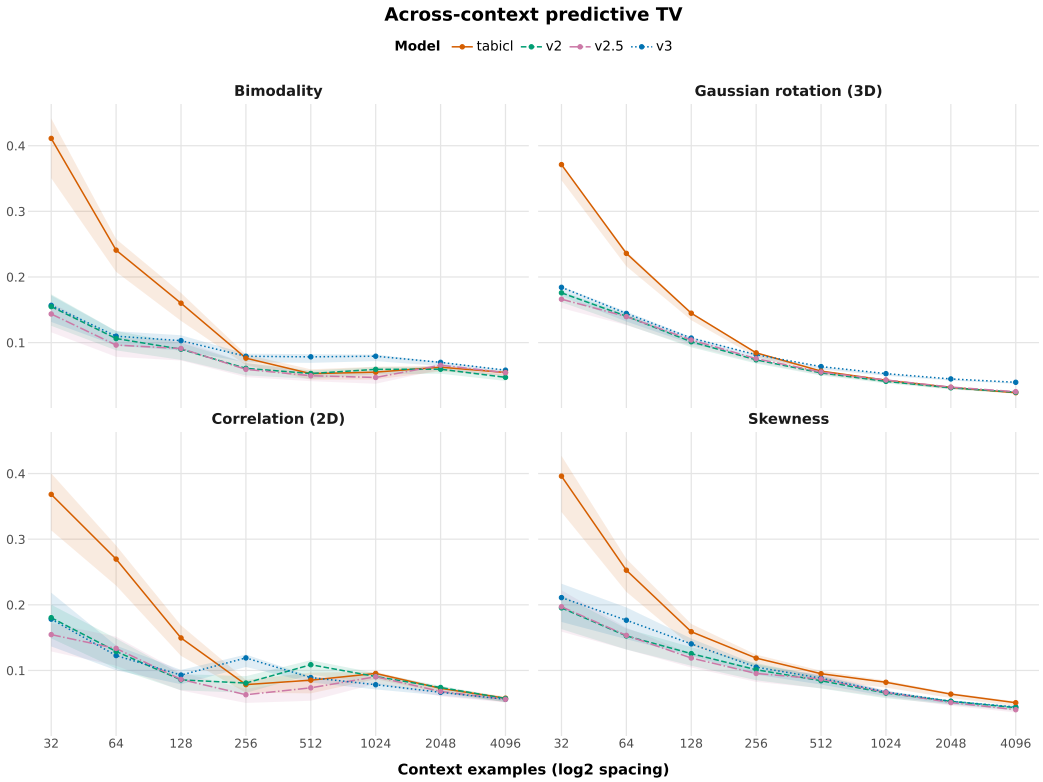

Figure 7: **Predictions depend less on which examples are drawn as context grows.** Mean total-variation distance between predictive distributions from pairs of independent contexts at the same setting and query, from 32 to 4,096 context rows.



Figure 8: **Native steering across context sizes in V2, V2.5, V3, and TabICL.** Patched minus unpatched TV error to the requested target posterior (negative is better) for one-step requests under a single calibration frozen at 4,096 rows; the benefit of steering generally grows with context and tracks the task-source patch, while norm-matched random directions stay near zero.

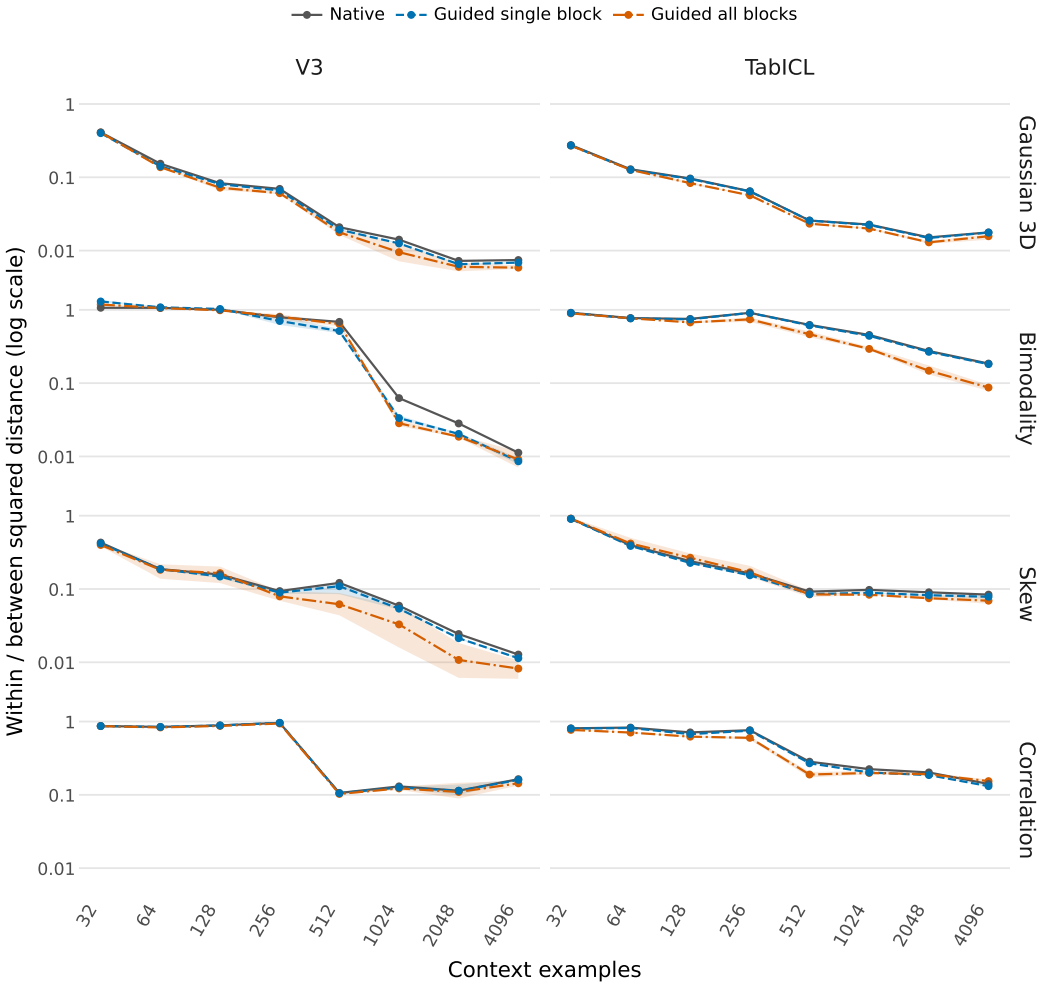

Figure 9: **Task representation separation before and after guided fine-tuning.** Within- to between-setting squared-distance ratio in the frozen native task basis (log scale; lower is better) for native TabPFN V3 and TabICL and for guided adapters trained at 512 rows; context length dominates, and guidance lowers the ratio only modestly, mostly with all-block updates.

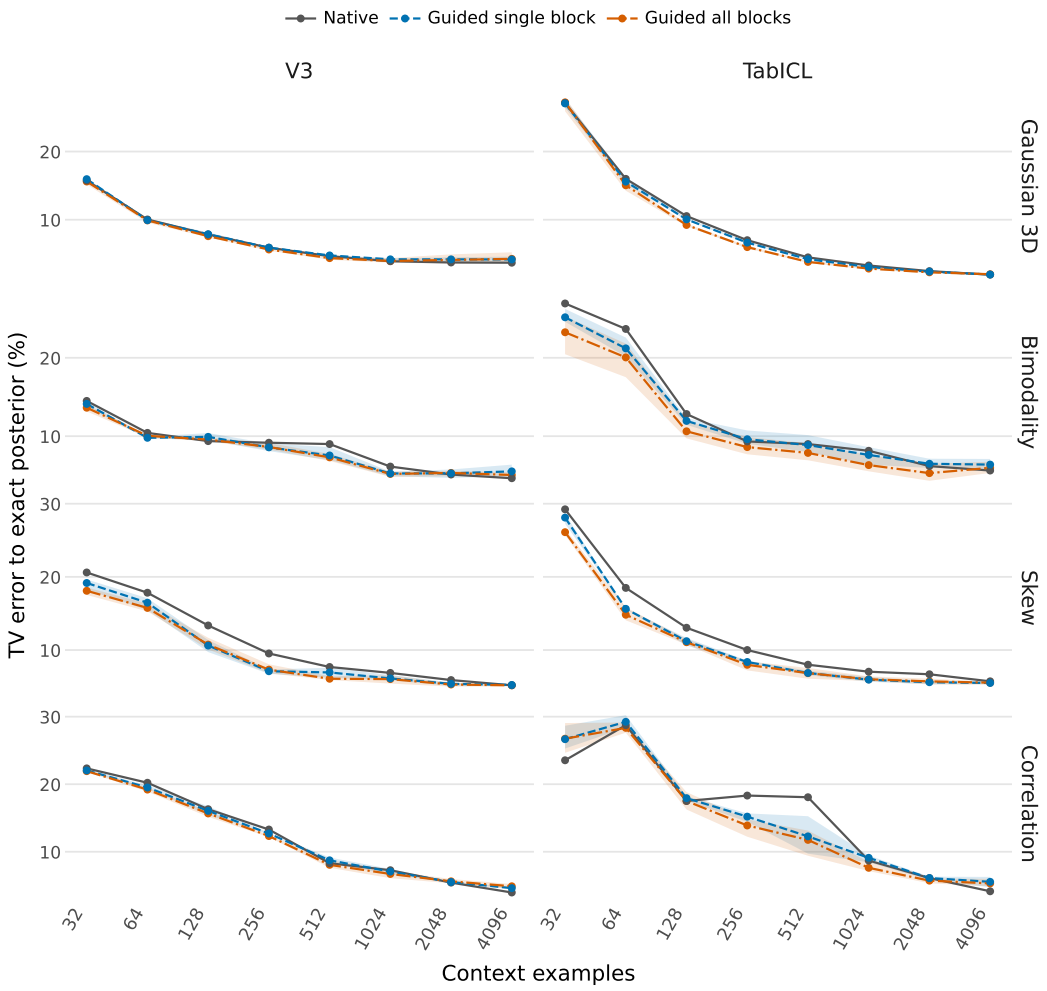

Figure 10: **Predictive accuracy before and after guided fine-tuning.** TV error to the exact posterior for the same checkpoints; guided adapters give modest and inconsistent gains, concentrated at short and intermediate context lengths, with shading spanning three fitted seeds.

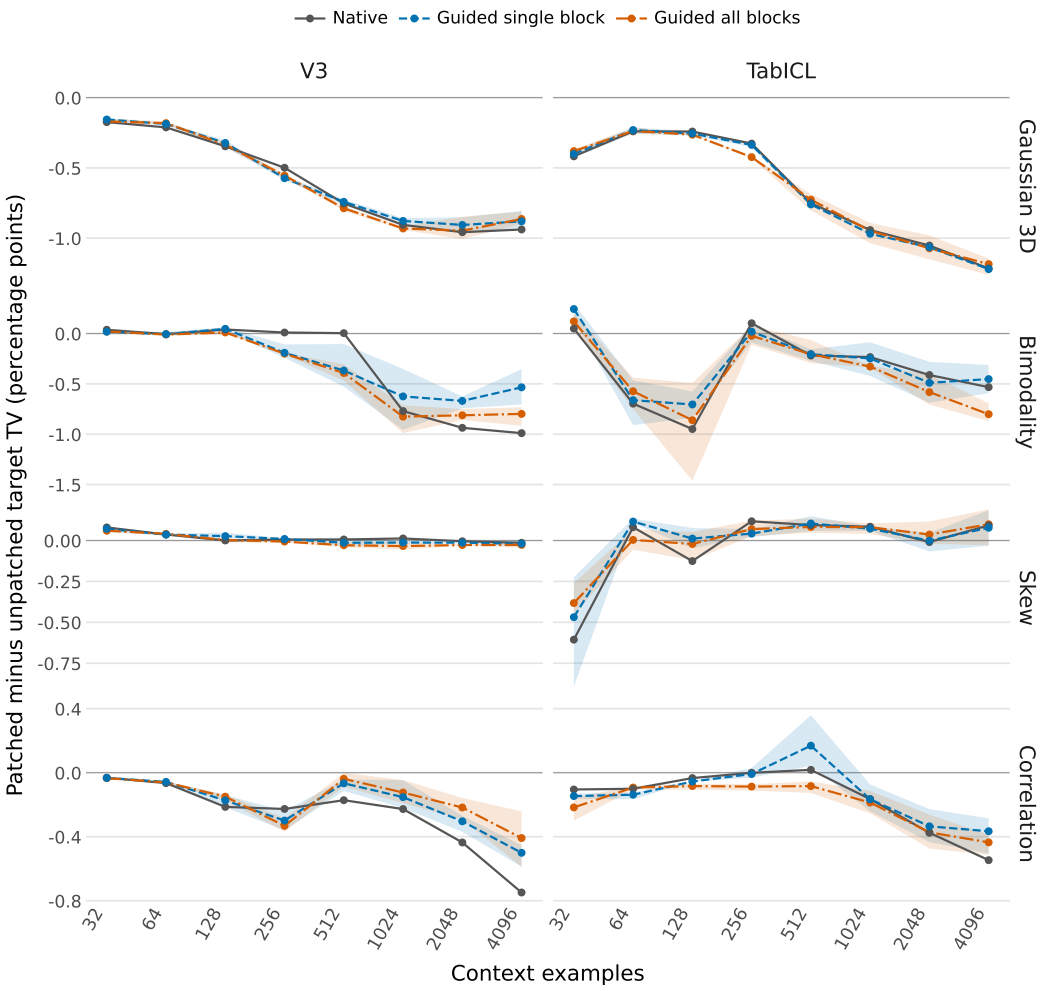

Figure 11: **Steering response before and after guided fine-tuning.** Patched minus unpatched TV error to the requested target when the same frozen native steering intervention is applied to each checkpoint; adapted models remain steerable and largely track the native response.

B.5 Extended Steering Results

Table 5 reports small-step movement and target-coordinate error with pointwise bootstrap intervals over twelve held-out contexts. Scalar responses are more heterogeneous than Gaussian ones; for example, V2.5 correlation moves slightly against the request at 512 rows. Spline interpolation of the calibration curve modestly reduces Gaussian target-coordinate error relative to piecewise-linear interpolation, without a consistent advantage across scalar tasks.

Table 5: Common small-step steering across TabPFN V2, V2.5, V3, and TabICL. Entries are paired steered-minus-identity means [95% pointwise context-bootstrap intervals] in requested-step units over 12 held-out contexts per cell. Positive desired shift and negative target error favor steering. Gaussian requests are $\pm 5^\circ$; scalar tasks use their declared natural-coordinate steps. Intervals are exploratory with no multiplicity adjustment.

Model	Task	Context rows	Desired shift	Target error
TabPFN V2	Gaussian 3D	512	+0.190 [+0.170, +0.208]	-0.178 [-0.204, -0.149]
TabPFN V2	Gaussian 3D	2,048	+0.295 [+0.279, +0.314]	-0.295 [-0.313, -0.279]
TabPFN V2	Bimodality	512	+0.277 [+0.070, +0.591]	+0.049 [-0.060, +0.218]
TabPFN V2	Bimodality	2,048	+0.553 [+0.382, +0.747]	-0.251 [-0.421, -0.100]
TabPFN V2	Skew	512	+0.299 [+0.139, +0.423]	-0.024 [-0.078, +0.043]
TabPFN V2	Skew	2,048	+0.480 [+0.352, +0.616]	-0.033 [-0.105, +0.026]
TabPFN V2	Correlation sweep	512	+0.195 [+0.124, +0.267]	0.000 [-0.035, +0.022]
TabPFN V2	Correlation sweep	2,048	+0.630 [+0.578, +0.680]	-0.208 [-0.278, -0.140]
TabPFN V2.5	Gaussian 3D	512	+0.066 [+0.058, +0.073]	-0.050 [-0.066, -0.034]
TabPFN V2.5	Gaussian 3D	2,048	+0.106 [+0.102, +0.110]	-0.106 [-0.110, -0.102]
TabPFN V2.5	Bimodality	512	+0.031 [0.000, +0.078]	-0.010 [-0.020, +0.001]
TabPFN V2.5	Bimodality	2,048	+0.206 [+0.161, +0.254]	-0.077 [-0.107, -0.047]
TabPFN V2.5	Skew	512	+0.272 [+0.208, +0.330]	-0.045 [-0.095, -0.007]
TabPFN V2.5	Skew	2,048	+0.421 [+0.328, +0.519]	-0.046 [-0.082, -0.010]
TabPFN V2.5	Correlation sweep	512	-0.019 [-0.024, -0.013]	-0.003 [-0.005, 0.000]
TabPFN V2.5	Correlation sweep	2,048	+0.519 [+0.484, +0.556]	-0.084 [-0.116, -0.049]
TabPFN V3	Gaussian 3D	512	+0.623 [+0.593, +0.652]	-0.569 [-0.614, -0.517]
TabPFN V3	Gaussian 3D	2,048	+0.743 [+0.730, +0.758]	-0.735 [-0.753, -0.718]
TabPFN V3	Bimodality	512	+0.184 [+0.070, +0.351]	-0.101 [-0.198, -0.035]
TabPFN V3	Bimodality	2,048	+0.377 [+0.323, +0.434]	-0.225 [-0.268, -0.180]
TabPFN V3	Skew	512	+0.487 [+0.382, +0.596]	-0.102 [-0.156, -0.049]
TabPFN V3	Skew	2,048	+0.669 [+0.624, +0.716]	-0.087 [-0.151, -0.031]
TabPFN V3	Correlation sweep	512	+0.682 [+0.633, +0.734]	-0.114 [-0.152, -0.071]
TabPFN V3	Correlation sweep	2,048	+0.783 [+0.756, +0.810]	-0.340 [-0.405, -0.274]
TabICL	Gaussian 3D	512	+0.679 [+0.644, +0.708]	-0.575 [-0.612, -0.531]
TabICL	Gaussian 3D	2,048	+0.833 [+0.818, +0.849]	-0.738 [-0.756, -0.718]
TabICL	Bimodality	512	+0.043 [+0.027, +0.058]	-0.037 [-0.059, -0.019]
TabICL	Bimodality	2,048	+0.502 [+0.351, +0.655]	-0.082 [-0.143, -0.024]
TabICL	Skew	512	+0.352 [+0.182, +0.521]	+0.045 [-0.051, +0.128]
TabICL	Skew	2,048	+0.691 [+0.535, +0.848]	-0.123 [-0.208, -0.029]
TabICL	Correlation sweep	512	+0.081 [+0.042, +0.126]	+0.021 [0.000, +0.040]
TabICL	Correlation sweep	2,048	+0.555 [+0.483, +0.621]	-0.063 [-0.096, -0.028]

B.6 Steering Control

These experiments ask how the Gaussian rotation steering response depends on the size of the request, the depth of the intervention, and which query tokens are patched.

B.6.1 How far can we steer?

Figures 13 and 14 compare requested and realized Gaussian rotation at depth approximately 0.75. At a 135° request with 2,048 rows, V3 and TabICL have negative mean signed angular movements, yet their conditional predictions move closer to their native predictions at the requested setting in squared Hellinger distance. This measures agreement with the model's own prediction, not accuracy

against the exact posterior. The nonmonotonic angular response therefore does not identify a sharp maximum steering distance.

B.6.2 Depthwise Steering

Each depth uses its own calibrated basis. Full target-state replacement at depth 0.75 yields $30.6°$ for the $67.5°$ request in V3, close to the $30.4°$ from steering, so the intermediate-depth shortfall persists even with the target state available. The common assay patches only feature tokens in V2 and V2.5, whereas V3 and TabICL expose a combined query-row state. In a matched V2.5 follow-up at depth approximately 0.75 and 2,048 rows, the same $67.5°$ request produces $2.28°$ with feature-only patches versus $31.69°$ with all query tokens. Apparent differences in steerability across models therefore partly reflect intervention scope (Figures 15 and 17).

B.7 Cross-Context Patching with a Shared Target Scale

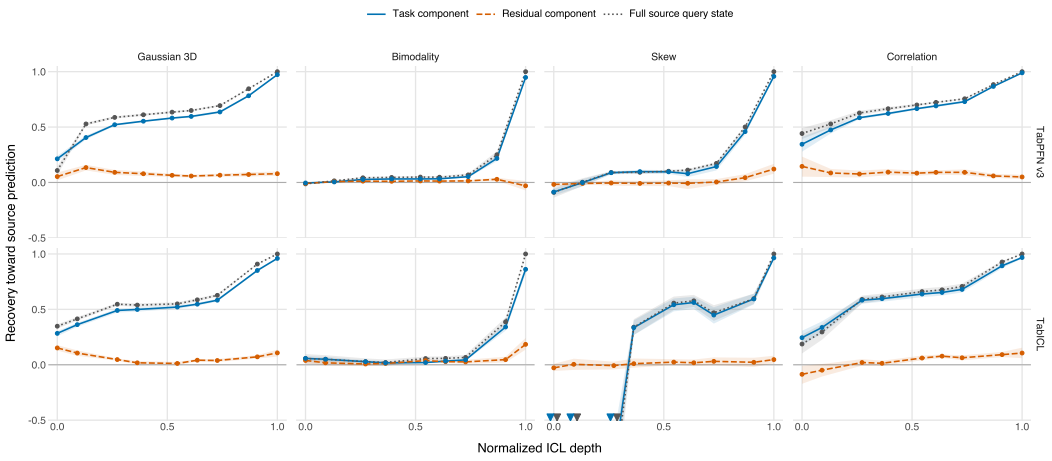

Figure 19: **Task components carry most of the late prediction change.** We patch query states between a source and a base context generated at different values of τ, keeping the query fixed. Blue copies the part of the source–base state difference inside the calibrated task subspace; orange copies the part outside it; gray copies the whole source query state. Each point comes from a fresh run with one patch after the selected block, with the base context in place. Recovery is the fraction of the squared Hellinger distance to the source prediction that the patch closes: one means matching the source, zero means no improvement, and negative values mean moving farther away. Contexts have 512 rows; eight calibration contexts fit the bases, and twelve other contexts evaluate four transfer pairs at nine depths. Bands are pointwise 95% intervals from 10,000 bootstrap draws of whole evaluation contexts, conditional on the checkpoint and calibration. Zero on the horizontal axis is after the first ICL block and one is after the last. Both models use a shared target scale with frozen weights. TabICL bimodality narrowly fails the native tracking criterion and is diagnostic only. Downward triangles mark the six estimates below -0.5, all task and full-state patches in the first three sampled TabICL skew blocks (lowest -2.84, at block zero).

All contexts and Gaussian heads use target mean zero and scale one for both the labels entering the model and output decoding. This keeps source and base states in the same target units. It changes the models' usual context-specific preprocessing while leaving their weights fixed. Baseline prediction error rises for TabICL: for example, mean squared Hellinger error against the exact skew distributions rises from 0.0022 to 0.0109. We rerun the original accuracy and shape checks without changing their thresholds. Seven of the eight model–task pairs pass. TabICL bimodality narrowly misses the central-mass tracking threshold, although all twelve evaluation contexts produce a bimodal endpoint. Its task component closes 86.1% of the final-block gap, but we treat it as diagnostic only.

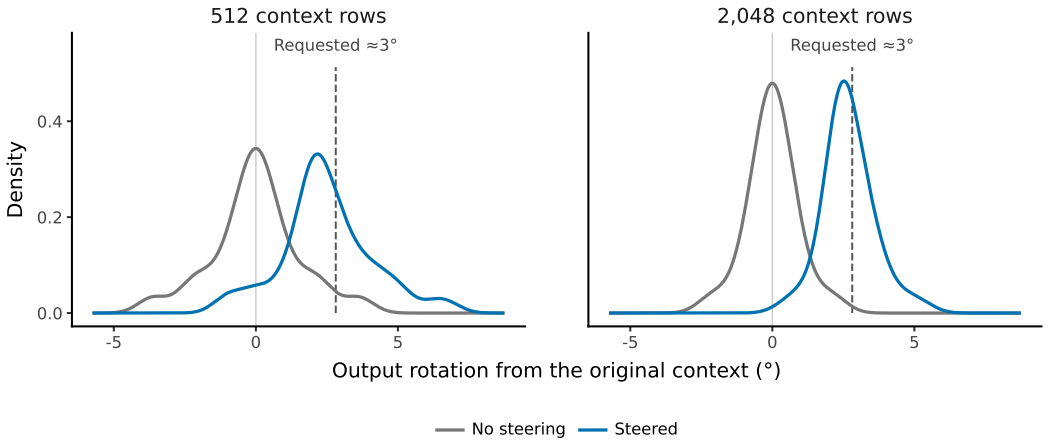

Figure 12: **Steered outputs follow a small requested rotation.** Densities of the decoded output rotation across held-out TabICL contexts for a 2.8125° Gaussian rotation request (dashed line): steered outputs (blue) center near the request while unsteered outputs (gray) center on the original phase, and both narrow from 512 to 2,048 context rows.

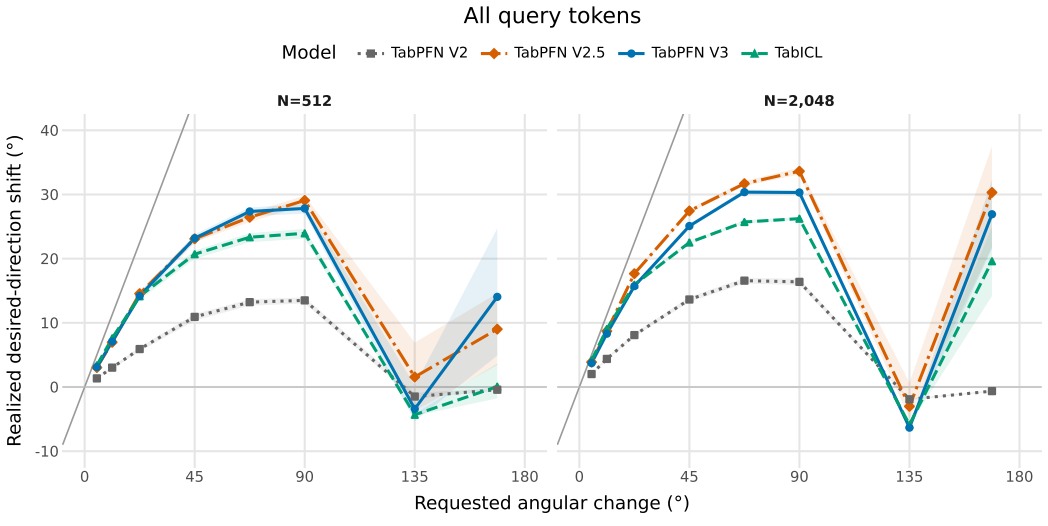

Figure 13: **Realized rotation falls increasingly short of larger requests.** Mean realized shift against requested Gaussian rotation when patching all query tokens at depth approximately 0.75, with the identity line in gray and pointwise 95% context-bootstrap bands; responses saturate below 35° near a 90° request, turn negative at 135°, and partly rebound.

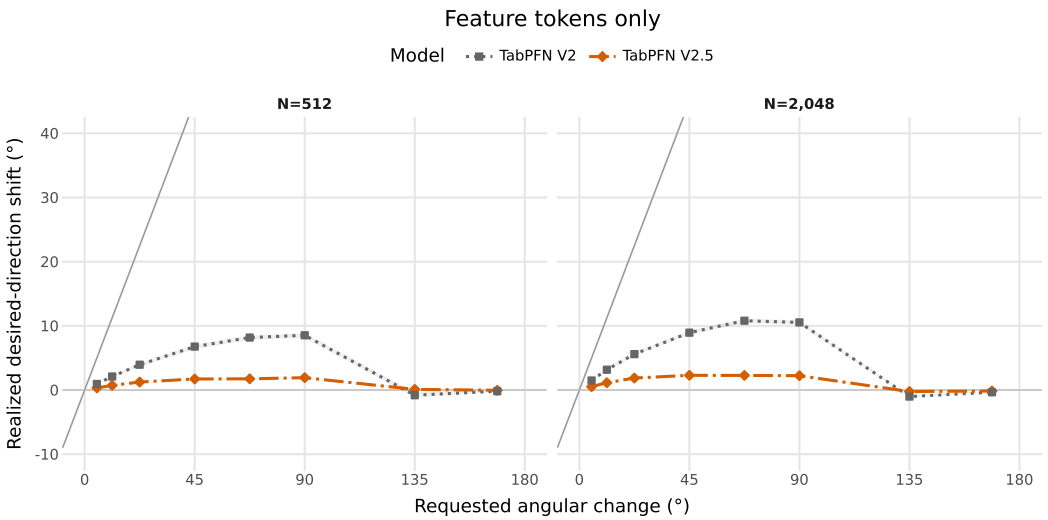

Figure 14: **Patching only feature tokens understates steerability.** The same sweep as Figure 13 with only feature tokens patched in TabPFN V2 and V2.5, which V3 and TabICL do not expose; V2 moves at most about 11° and V2.5 about 2°.

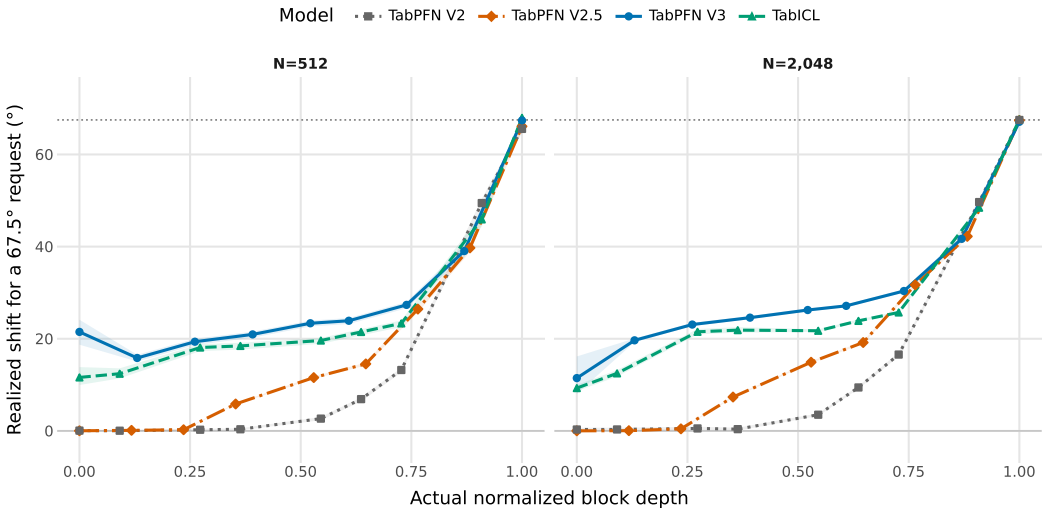

Figure 15: **Large requests are fully realized only at the final block.** Realized shift for a 67.5° Gaussian rotation request against actual normalized block depth, patching all query tokens with a separately calibrated basis at each depth; every model reaches the request (dotted line) only at the last block.

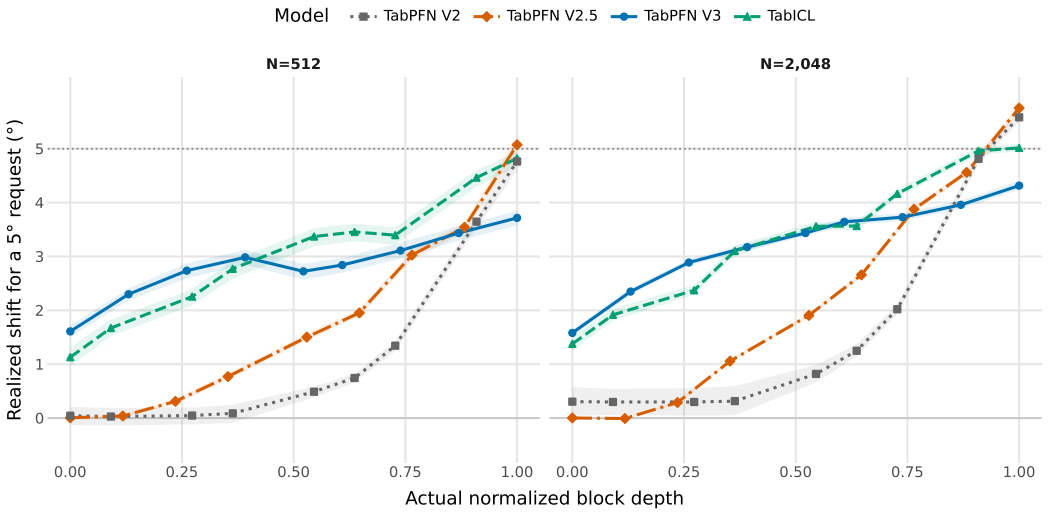

Figure 16: **Small requests are realized progressively with depth.** The same depth profile for a 5° request with all query tokens patched; TabPFN V3 and TabICL realize over half of the request from mid-depth onward, whereas V2 and V2.5 respond mostly in later blocks.

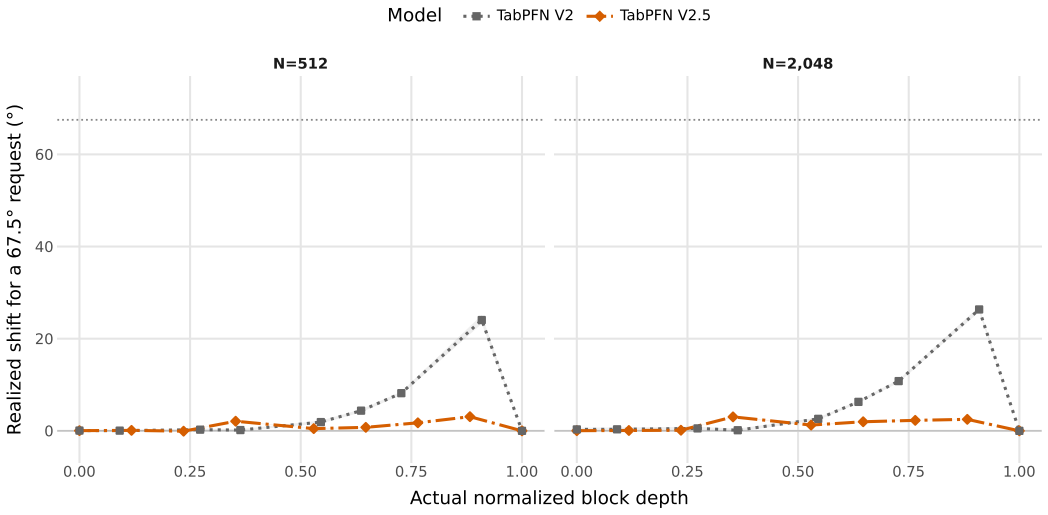

Figure 17: **Feature-token patches fall far short of the request at every depth.** Depth profile for the 67.5° request in TabPFN V2 and V2.5 with only feature tokens patched; V2 peaks at about 25° just before the final block, V2.5 stays near zero, and neither responds at the final block.

Table 6: Update depth and held-out prediction error. Rows and fits match Table 1: 192 fits, three fitted seeds per condition, eight held-out contexts and unseen task settings per fit. Entries are mean seedwise percentage changes in Hellinger-squared error to the exact posterior, relative to each fit's native-512 baseline: 100 times (mean adapted error / mean native error - 1), then averaged over seeds. Negative values indicate lower error. Gaussian 3D uses the original moment-matched joint-Gaussian reconstruction; the other tasks use binned predictive distributions, averaging three anchor queries for correlation. These are descriptive comparisons within each model and task; Gaussian values do not measure direct conditional-bin error. Quarter- and half-depth fits use the original B200 MIG run; full-B200 repeats are excluded. Block indices are zero-based, with requested relative depth in parentheses. All-block updates have more trainable parameters.

			Change in prediction error, Hellinger^2 (%)			
Model	Objective	Updated blocks	Gaussian 3D	Bimodality	Skew	Correlation
TabPFN V3	Output-only	6 (0.25)	+257.0	−49.2	−19.7	+9.1
		12 (0.50)	+793.2	−53.1	−14.6	−2.3
		17 (0.75)	+1001.5	−57.8	−20.1	−9.1
		All ICL blocks	+1261.2	−63.6	−22.8	−21.8
	Guided	6 (0.25)	−6.7	−53.9	−20.8	0.0
		12 (0.50)	−7.3	−52.1	−16.3	−4.5
		17 (0.75)	+403.8	−57.4	−24.7	−7.1
		All ICL blocks	+1283.7	−63.0	−26.7	−20.9
TabICL	Output-only	3 (0.25)	−12.7	−10.2	−28.5	−62.7
		6 (0.50)	−16.8	−20.6	−37.6	−66.1
		8 (0.75)	−10.5	−24.0	−38.1	−59.1
		All ICL blocks	−15.0	−43.3	−41.2	−69.8
	Guided	3 (0.25)	−9.6	−8.5	−26.8	−60.5
		6 (0.50)	−11.1	−10.2	−31.8	−64.5
		8 (0.75)	−8.4	−18.7	−32.9	−57.0
		All ICL blocks	−13.3	−42.0	−34.2	−69.1

For bimodality, skew, and correlation, we draw observations from a fixed distribution, then draw targets from the known conditional distribution for each mechanism setting. These are new joint distributions that preserve the posterior families at the fixed queries, and differ from the scalar tasks in the other experiments. Correlation uses three fixed conditioning values, -0.9, 0, and 0.9. The Gaussian rotation retains its original simulator and combines query states from three conditional heads. Each layer has a separately calibrated basis. Identity and replay checks pass at all 72 evaluated model–task–depth combinations, and patching the full source state at the final block recovers the source prediction within numerical tolerance in every pair; this is an implementation check.

B.8 Fine-Tuning

This section details the long-to-short adaptation experiment of Section 6 and reports its complete results.

B.8.1 Method Overview

Figure 20 summarizes the training setup, in which a LoRA-adapted student given a 512-row prefix is trained against a frozen teacher given the full 2,048-row context.

B.8.2 Complete Results

Table 6 reports held-out prediction error for the same 192 fits as Table 1. Figures 21 and 22 come from an earlier six-seed cohort, summarized for Gaussian rotation, bimodality, and skew in Tables 9 and 10; Tables 7 and 8 report the current three-seed comparison across four tasks and two update scopes.

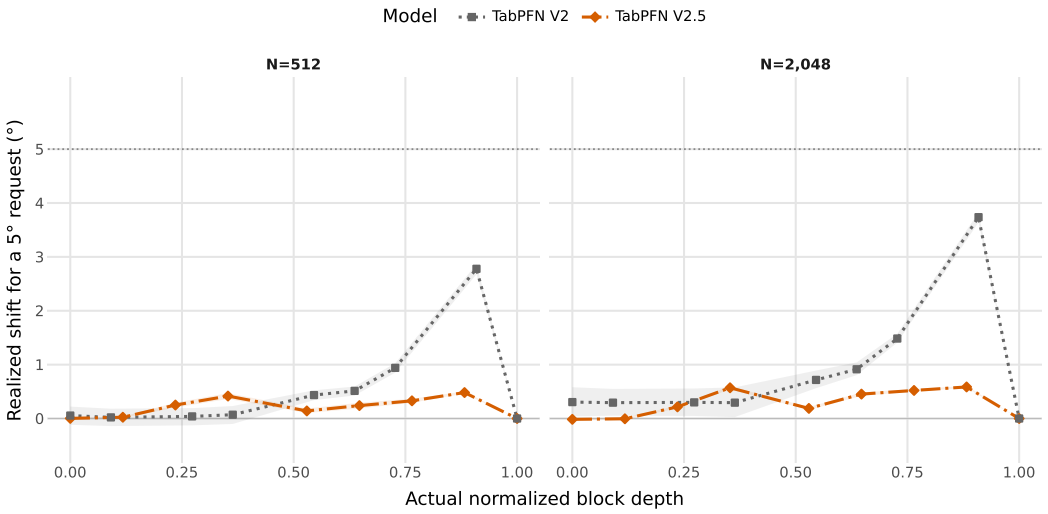

Figure 18: **The feature-token shortfall persists for small requests.** Depth profile for the $5°$ request in TabPFN V2 and V2.5 with only feature tokens patched; V2 peaks below $4°$ and V2.5 below $1°$, again with no response at the final block.

Table 7: Guided adaptation across four tasks and two update scopes. Each condition uses three fitted seeds from the scope experiment (48 fits total). Entries are means; brackets give observed fitted-seed minima and maxima, not confidence intervals. Representation columns are seedwise percentage changes from native 512 on eight held-out contexts, using the frozen task basis. TV columns use the same adapters on 16 fresh paired contexts and 16 unseen settings per fit, as in Table 2. TV error is distance to the exact posterior; N–A TV is prediction movement from native to adapted. TV uses a common physical grid with overflow bins, equal weights for Gaussian autoregressive heads after averaging their anchors, and three anchors for correlation. Late updates modify V3 block 17 or TabICL block 8 (zero-based); readouts stay at blocks 17 and 11. All-ICL updates have more trainable parameters. The older six-seed cohort is retained in Table 9.

Model	Task	Updated blocks	Variance Δ (%)	Distance Δ (%)	TV error N→A (%)	Δ TV (pp)	N–A TV (pp)
TabPFN V3	Gaussian 3D	Late (17)	-3.0 [-8.0, 0.5]	-11.9 [-17.6, -7.8]	4.86 → 5.12	0.26 [0.18, 0.31]	3.36 [3.24, 3.59]
		All ICL	-13.8 [-19.0, -8.7]	-19.2 [-23.2, -17.1]	4.86 → 4.79	-0.07 [-0.20, 0.14]	3.94 [3.62, 4.20]
	Bimodality	Late (17)	17.9 [10.0, 26.2]	-24.6 [-29.8, -17.7]	7.88 → 6.12	-1.75 [-1.89, -1.61]	5.07 [4.75, 5.24]
		All ICL	-32.6 [-39.8, -22.7]	-52.9 [-57.7, -48.7]	7.88 → 5.84	-2.03 [-2.09, -1.95]	4.99 [4.58, 5.50]
	Skew	Late (17)	-4.3 [-5.4, -3.2]	-9.9 [-12.4, -8.6]	6.20 → 5.26	-0.94 [-1.15, -0.80]	3.10 [2.61, 3.43]
		All ICL	-47.0 [-50.7, -43.7]	-31.5 [-36.7, -22.9]	6.20 → 5.21	-0.99 [-1.37, -0.79]	3.09 [2.96, 3.32]
	Correlation	Late (17)	-6.3 [-10.0, -3.6]	-2.5 [-4.5, -1.0]	7.12 → 6.65	-0.47 [-0.69, -0.20]	3.17 [2.87, 3.48]
		All ICL	-27.9 [-34.7, -22.6]	-3.8 [-5.4, -0.7]	7.12 → 6.23	-0.90 [-1.13, -0.65]	3.17 [2.99, 3.36]
TabICL	Gaussian 3D	Late (8)	-19.3 [-24.5, -13.8]	-42.7 [-49.9, -38.7]	4.60 → 4.33	-0.27 [-0.40, -0.14]	1.21 [0.99, 1.36]
		All ICL	-28.2 [-33.2, -22.9]	-48.0 [-48.6, -47.6]	4.60 → 4.05	-0.55 [-0.65, -0.44]	1.80 [1.77, 1.87]
	Bimodality	Late (8)	-3.5 [-6.6, 1.1]	-10.5 [-12.7, -9.3]	7.13 → 6.94	-0.19 [-0.28, -0.07]	2.69 [2.11, 3.69]
		All ICL	-23.2 [-28.5, -14.6]	-23.5 [-26.5, -17.5]	7.13 → 6.02	-1.10 [-1.23, -0.91]	3.61 [3.36, 3.86]
	Skew	Late (8)	-14.0 [-20.4, -10.6]	-6.4 [-9.4, -1.4]	6.43 → 5.32	-1.11 [-1.24, -0.98]	2.80 [2.57, 3.02]
		All ICL	-31.9 [-46.6, -19.5]	-19.9 [-25.4, -13.3]	6.43 → 5.20	-1.23 [-1.40, -1.09]	3.17 [2.92, 3.61]
	Correlation	Late (8)	54.8 [41.3, 62.5]	-43.3 [-45.9, -40.7]	13.73 → 9.58	-4.14 [-4.87, -3.06]	10.85 [10.19, 11.32]
		All ICL	64.2 [25.6, 102.6]	-64.8 [-67.5, -60.9]	13.73 → 8.04	-5.69 [-6.33, -4.84]	11.39 [11.18, 11.79]

Table 8: Hellinger-squared error for the same four-task, two-scope guided adapters as Table 7. Native and adapted columns are mean absolute errors to the exact posterior; Δ is adapted minus native 512, paired within each fit. Each condition has three fitted seeds and eight original held-out contexts per fit with unseen settings. Brackets give observed seed minima and maxima, not confidence intervals. Gaussian 3D uses the original moment-matched joint-Gaussian reconstruction; other tasks use binned predictive distributions, averaging three anchor queries for correlation. The primary table's TV uses direct conditional bins and 16 fresh contexts, so these columns differ in both diagnostic and evaluation sample. Late updates modify V3 block 17 or TabICL block 8; all-ICL updates have more trainable parameters. The historical six-seed analysis is retained in Table 10.

Model	Task	Updated blocks	Native Hellinger^2	Adapted Hellinger^2	Δ Hellinger^2
TabPFN V3	Gaussian 3D	Late (17)	0.00388	0.01906	0.01517 [0.01069, 0.02011]
		All ICL	0.00388	0.05390	0.05001 [0.04169, 0.05958]
	Bimodality	Late (17)	0.00708	0.00302	-0.00406 [-0.00470, -0.00370]
		All ICL	0.00708	0.00262	-0.00446 [-0.00490, -0.00418]
	Skew	Late (17)	0.00483	0.00350	-0.00133 [-0.00276, -0.00037]
		All ICL	0.00483	0.00344	-0.00139 [-0.00253, -0.00071]
	Correlation	Late (17)	0.00527	0.00490	-0.00037 [-0.00076, -0.00016]
		All ICL	0.00527	0.00416	-0.00110 [-0.00158, -0.00085]
TabICL	Gaussian 3D	Late (8)	0.00438	0.00400	-0.00038 [-0.00049, -0.00019]
		All ICL	0.00438	0.00382	-0.00056 [-0.00074, -0.00035]
	Bimodality	Late (8)	0.00852	0.00693	-0.00159 [-0.00215, -0.00102]
		All ICL	0.00852	0.00495	-0.00358 [-0.00385, -0.00319]
	Skew	Late (8)	0.00491	0.00332	-0.00158 [-0.00196, -0.00138]
		All ICL	0.00491	0.00327	-0.00164 [-0.00176, -0.00149]
	Correlation	Late (8)	0.02757	0.01168	-0.01588 [-0.01945, -0.01113]
		All ICL	0.02757	0.00845	-0.01911 [-0.02178, -0.01570]

Table 9: Historical six-seed guided-adaptation results for three tasks. Representation columns are seedwise changes relative to native short context. TV columns report predictive error to the exact posterior and direct native-to-adapted prediction movement. Gaussian TV averages the three autoregressive heads and their fixed anchor queries; scalar TV uses the frozen posterior bins. Each model retains its frozen output grid, so cross-model absolute levels are descriptive and the paired within-cell changes are primary. TV values are percentage points. Brackets are exact pointwise 95% percentile bootstrap intervals over all 46,656 six-of-six seed resamples. The current four-task scope comparison is in Table 7.

Model	Task	Variance Δ (%)	Distance Δ (%)	TV error N$\rightarrow$A (%)	Δ TV (pp)	N–A TV (pp)
TabPFN V3	Gaussian 3D	-4.4 [-9.4, -0.8]	-13.1 [-16.4, -9.7]	10.6 $\rightarrow$ 8.6	-2.0 [-2.1, -1.9]	6.2 [6.1, 6.3]
TabPFN V3	Bimodality	16.9 [6.6, 28.8]	-29.5 [-33.6, -25.5]	7.0 $\rightarrow$ 5.1	-1.8 [-2.2, -1.5]	4.9 [4.4, 5.5]
TabPFN V3	Skew	-10.0 [-12.7, -7.7]	-10.7 [-12.7, -8.8]	6.1 $\rightarrow$ 4.9	-1.2 [-1.7, -0.7]	3.3 [2.8, 3.7]
TabICL	Gaussian 3D	-12.7 [-15.9, -10.2]	-39.3 [-42.7, -34.8]	4.5 $\rightarrow$ 4.3	-0.1 [-0.2, -0.1]	1.1 [1.0, 1.3]
TabICL	Bimodality	1.8 [0.2, 3.5]	-11.2 [-17.5, -5.4]	7.1 $\rightarrow$ 6.6	-0.5 [-0.9, -0.2]	2.3 [1.3, 3.0]
TabICL	Skew	-0.8 [-1.8, 0.2]	-2.9 [-4.6, -1.3]	6.6 $\rightarrow$ 5.4	-1.2 [-1.4, -1.0]	2.8 [2.3, 3.1]

Table 10: Historical six-seed Hellinger-squared analysis. Native and adapted columns are absolute error to the exact posterior; Δ is adapted minus native short. Scalar cells use binned posterior Hellinger^2. Gaussian cells use the prespecified moment-matched joint-Gaussian reconstruction, whereas Table 9 reports direct conditional-bin TV for these historical six-seed adapters. The V3 Gaussian disagreement therefore localizes the regression to reconstructed joint structure rather than its fixed-anchor conditional predictions. Brackets are exact pointwise 95% percentile bootstrap intervals. The current four-task scope comparison is in Table 8.

Model	Task	Native Hellinger^2	Adapted Hellinger^2	Δ Hellinger^2
TabPFN V3	Gaussian 3D	0.00358	0.02025	0.01667 [0.01211, 0.02130]
TabPFN V3	Bimodality	0.00716	0.00329	-0.00388 [-0.00428, -0.00348]
TabPFN V3	Skew	0.00459	0.00299	-0.00160 [-0.00219, -0.00097]
TabICL	Gaussian 3D	0.00399	0.00386	-0.00013 [-0.00024, -0.00004]
TabICL	Bimodality	0.00822	0.00665	-0.00157 [-0.00229, -0.00076]
TabICL	Skew	0.00512	0.00365	-0.00147 [-0.00173, -0.00119]

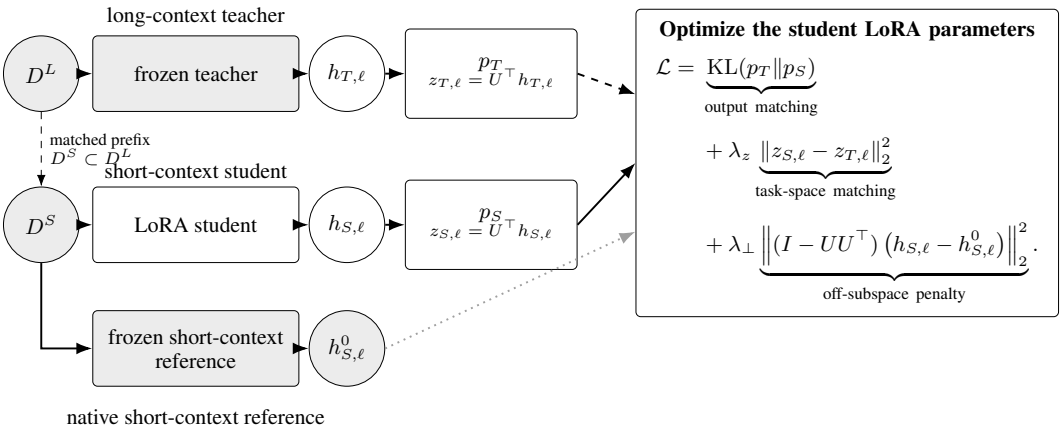

Figure 20: **Long-to-short representation-guided self-distillation.** A frozen teacher processes a long context D^L, while a LoRA-adapted student processes its matched short prefix D^S. Both receive the same query, which is omitted for visual clarity. The student is trained to match the teacher's predictive distribution p_T and task coordinates $z_{T,\ell} = U^\top h_{T,\ell}$. An off-subspace penalty limits changes outside the identified task subspace relative to the native short-context state $h^0_{S,\ell}$. Only the student's LoRA parameters are updated.

B.8.3 Dense Hidden Vector Guidance

Table 11 adds two dense-guidance objectives to the depth and scope comparison. Both use a single-block rank-eight LoRA update at V3 block 17 or TabICL block 8.

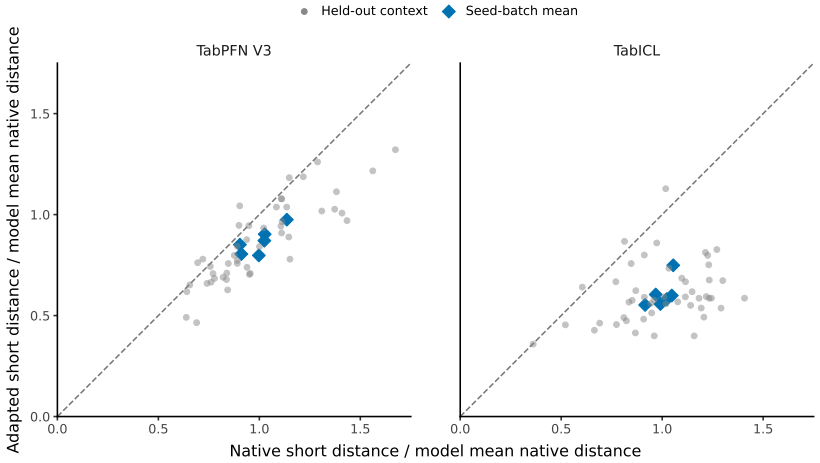

Figure 21: **Guided fine-tuning moves short-context states toward their long-context targets.** Squared task-coordinate distance to the matched 2,048-row state before (horizontal) and after (vertical) guided adaptation on Gaussian rotation, normalized by each model's mean native distance, for 48 held-out contexts (gray) and six seed means (blue); points below the diagonal are closer, and mean distance falls by 13.3% for TabPFN V3 and 39.3% for TabICL.

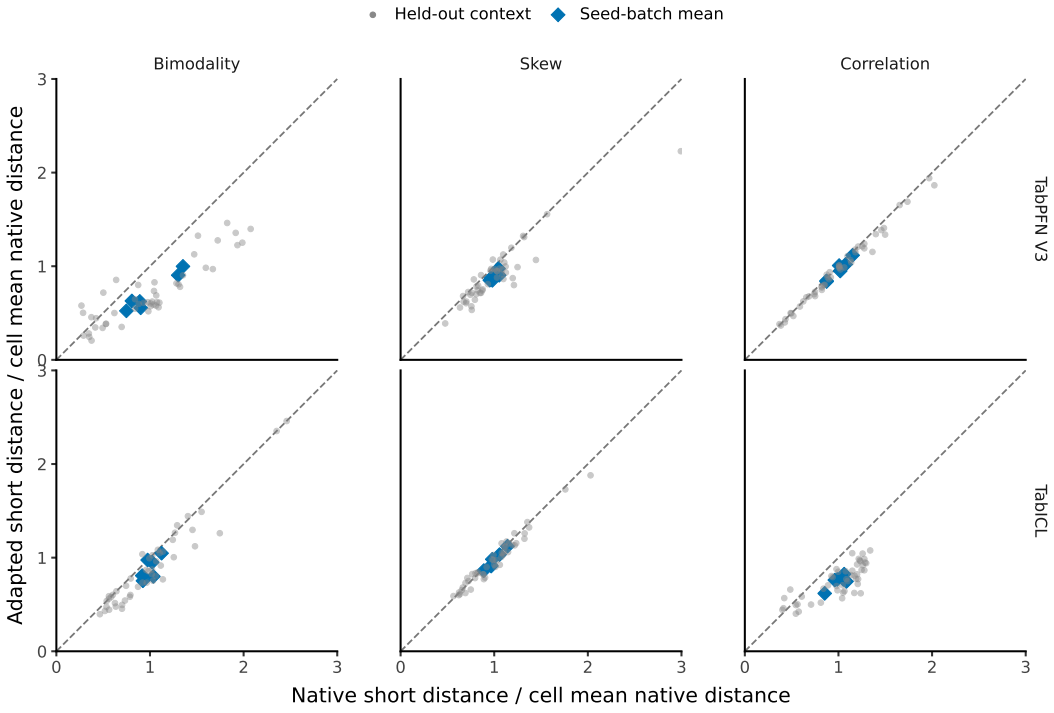

Figure 22: **Guided matching extends unevenly to the scalar tasks.** The same comparison as Figure 21 for bimodality, skew, and correlation in TabPFN V3 (top) and TabICL (bottom); mean distance falls most for V3 bimodality (29%) and TabICL correlation (25%), and by 3–11% elsewhere.

Table 11: Extended adaptation-distance comparison, including dense guidance. The 16 depth/scope rows repeat Table 1; four new rows add 48 B200 MIG dense fits (240 fits total, three seeds per condition). Entries are mean seedwise percentage changes in squared task-space distance to the paired native-2048 state relative to native 512; negative values indicate closer matching. Each seed uses eight held-out contexts and unseen task settings. Dense guidance matches the full hidden vector during training; all rows are evaluated in the same frozen task basis. Dense + preserve retains the task-guided objective's orthogonal-short-state penalty; Dense omits it. Dense fits update only V3 block 17 or TabICL block 8, with fixed readouts at blocks 17 and 11. Block indices are zero-based; parentheses give requested relative depth. All-block updates have more trainable parameters.

			Change in distance to long context (%)			
Model	Objective	Updated blocks	Gaussian 3D	Bimodality	Skew	Correlation
TabPFN V3	Output-only	6 (0.25)	+451.3	−15.2	+59.3	+4.7
		12 (0.50)	+416.4	+60.7	+59.0	+4.2
		17 (0.75)	+307.7	+7.1	+20.7	−1.1
		All ICL blocks	+41.5	−22.1	+22.4	+1.4
	Task-guided	6 (0.25)	−14.8	−45.9	−21.5	+0.4
		12 (0.50)	−16.8	−47.6	−20.7	−0.02
		17 (0.75)	−11.9	−24.6	−9.9	−2.5
		All ICL blocks	−19.2	−52.9	−31.5	−3.8
	Dense	17 (0.75)	+7.5	−14.8	−1.4	−1.6
	Dense + preserve	17 (0.75)	+2.8	−15.9	−2.6	−2.1
TabICL	Output-only	3 (0.25)	−2.5	−8.2	+0.7	−41.1
		6 (0.50)	+20.7	−7.5	+14.2	−43.1
		8 (0.75)	+105.2	+6.0	+21.8	−36.4
		All ICL blocks	+125.6	−4.9	+20.6	−55.7
	Task-guided	3 (0.25)	−44.1	−11.8	−18.2	−51.1
		6 (0.50)	−45.2	−9.3	−18.4	−51.5
		8 (0.75)	−42.7	−10.5	−6.4	−43.3
		All ICL blocks	−48.0	−23.5	−19.9	−64.8
	Dense	8 (0.75)	−43.4	−4.9	−1.1	−42.9
	Dense + preserve	8 (0.75)	−43.0	−3.6	−2.0	−40.9

B.8.4 STEERING POST-FINE-TUNING

Table 12 checks whether adaptation changes how the model responds to steering. Every adapted checkpoint remains steerable under the native 2,048-row calibration, and tail error generally falls toward the native 2,048-row level, though guidance does not reduce it beyond output-only adaptation.

Table 12: Post-fine-tuning stability and steering. Entries are descriptive means over three fitted seeds, with 16 fresh evaluation contexts per seed. Tail MSE averages the lower and upper 5% tails. Steering averages the two one-step spline interventions. ΔTV is steered minus unsteered TV to the requested target; response error is absolute error in the realized shift, divided by the requested step (5° for Gaussian 3D; 0.05 otherwise). Lower is better in every column.

	TabPFN V3			TabICL		
Condition	Tail MSE ($\times 10^4$)	ΔTV ($\times 10^3$)	Resp. error (steps)	Tail MSE ($\times 10^4$)	ΔTV ($\times 10^3$)	Resp. error (steps)
Gaussian 3D						
Native 512	1.70	-7.417	0.203	2.36	-8.254	0.093
Native 2048	0.85	-9.684	0.213	0.67	-12.890	0.082
Single block, output-only	1.63	-7.252	0.167	1.85	-9.081	0.083
Single block, guided	1.77	-7.123	0.191	2.05	-8.585	0.081
All ICL blocks, output-only	1.50	-7.493	0.202	1.81	-9.461	0.083
All ICL blocks, guided	1.55	-7.663	0.196	1.85	-8.902	0.081
Bimodality						
Native 512	1.96	-0.837	0.874	2.11	-0.209	1.156
Native 2048	0.88	-3.762	0.687	0.68	-0.545	0.806
Single block, output-only	1.66	-2.700	0.750	1.54	-0.328	0.803
Single block, guided	1.80	-2.703	0.794	1.71	-0.172	0.762
All ICL blocks, output-only	1.51	-2.705	0.749	1.40	-0.620	0.822
All ICL blocks, guided	1.57	-3.164	0.761	1.77	-0.746	0.807
Skew						
Native 512	2.63	-0.013	0.420	3.73	-0.088	0.428
Native 2048	1.26	-0.120	0.317	1.49	-0.110	0.416
Single block, output-only	1.98	-0.048	0.446	2.53	-0.117	0.438
Single block, guided	1.88	-0.066	0.440	2.81	-0.116	0.431
All ICL blocks, output-only	1.98	-0.114	0.392	2.46	-0.108	0.462
All ICL blocks, guided	1.90	-0.073	0.384	2.61	-0.145	0.423
Correlation						
Native 512	7.39	-2.501	0.175	75.54	-0.341	0.169
Native 2048	2.04	-3.571	0.137	4.40	-3.104	0.233
Single block, output-only	4.89	-2.576	0.181	14.92	-1.963	0.223
Single block, guided	4.89	-2.526	0.186	17.79	-1.812	0.216
All ICL blocks, output-only	4.30	-2.715	0.157	9.79	-2.191	0.223
All ICL blocks, guided	4.26	-2.785	0.165	11.69	-2.143	0.225

All adapted models use 512 context rows. All-block updates apply LoRA across ICL blocks, with a larger parameter budget than single-block updates. Native rows use the single-block/output-only evaluation replicate. The native-2048 steering calibration is frozen within each evaluation; direction averages can hide asymmetric responses.

B.8.5 RANK SENSITIVITY

Remeasuring the six-seed adapters of Table 9 in seven rather than eight task coordinates leaves the sign of every change unchanged (Table 13). Magnitudes are also similar, except that the TabICL Gaussian rotation distance reduction shrinks from 39% to 29%.

Table 13: Post hoc seven-coordinate measurement sensitivity for the five rank-eight cells in the historical Table 9. Entries are means of within-seed percentage changes from native short context. The rank-seven columns remeasure saved states from the original fitted adapters; they are not rank-seven training or causal-intervention results.

		Context variance Δ (%)		Distance Δ (%)	
Model	Task	Rank 8	Rank 7	Rank 8	Rank 7
TabPFN V3	Bimodality	16.89	19.09	-29.50	-31.24
TabPFN V3	Skew	-10.04	-9.92	-10.74	-11.25
TabICL	Gaussian 3D	-12.68	-10.05	-39.32	-29.38
TabICL	Bimodality	1.85	1.65	-11.20	-12.23
TabICL	Skew	-0.76	-0.84	-2.95	-3.40

B.9 Representations Across Models and Tasks

The following figures extend the Gaussian rotation display of Figure 2 to the three scalar tasks. Because these properties are not periodic, states trace open paths ordered by the task setting.

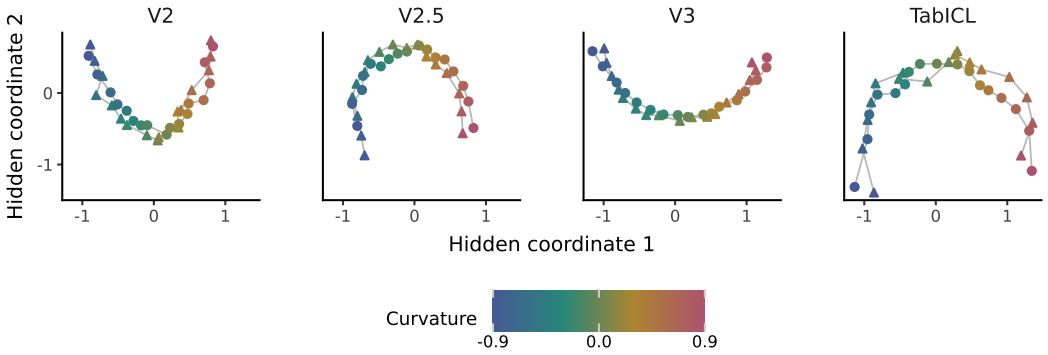

Figure 23: **Skew traces an ordered, open path in every model.** Native query states at depth 0.75 for 17 curvature settings in two held-out 512-row contexts (circles and triangles), projected onto a two-dimensional PCA fitted on a separate context.

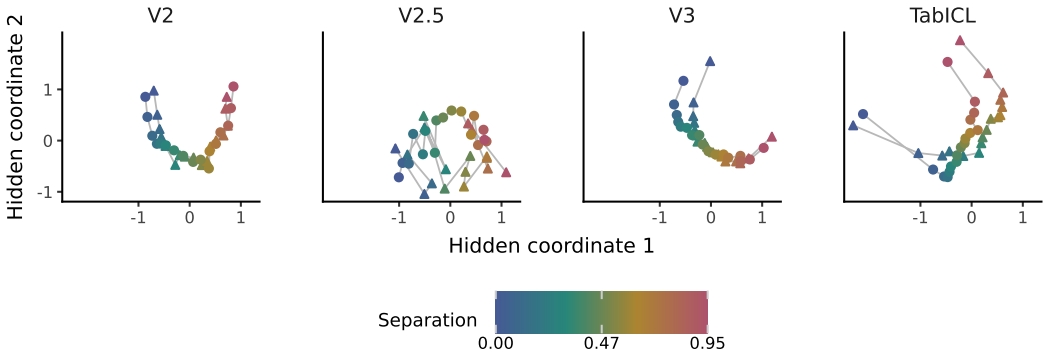

Figure 24: **Bimodality paths agree less across contexts.** Native query states for 17 mode-separation settings, with the same contexts and projection protocol as Figure 23. The two contexts diverge most for V2.5 and TabICL, consistent with the slow identification of bimodality in Figure 5.

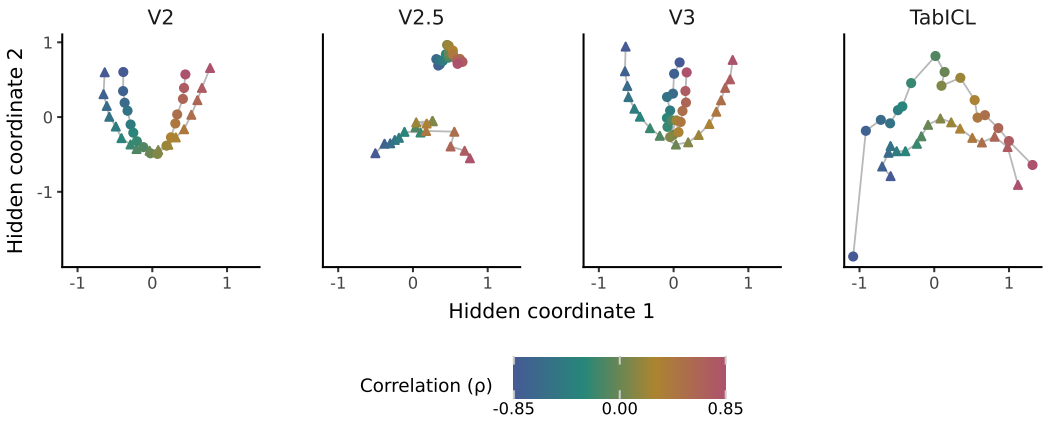

Figure 25: **Correlation paths are ordered but offset between contexts.** Native query states for 17 settings of ρ from -0.85 to 0.85, with the same protocol as Figure 23. Each context traces its own ordered path, but the two paths are displaced from one another.