# Task Vector Geometry Underlies Dual Modes of Task Inference in Transformers*

**Hao Yan**[†] **Haolin Yang**[‡] **Yiqiao Zhong**[†]
[†]Department of Statistics, University of Wisconsin–Madison
[‡]Center for Data Science, New York University
hyan84@wisc.edu hy3127@nyu.edu yiqiao.zhong@wisc.edu

## Abstract

Transformers are effective at inferring the latent task from context via two inference modes: recognizing a task seen during training, and adapting to a novel one. Recent interpretability studies have identified from middle-layer representations task-specific directions, or *task vectors*, that steer model behavior. However, a lack of rigorous foundations hinders connecting internal representations to external model behavior: existing work fails to explain how task-vector geometry is shaped by the training distribution, and what geometry enables out-of-distribution (OOD) generalization. In this paper, we study these questions in a controlled synthetic setting by training small transformers from scratch on latent-task sequence distributions, which allows a principled mathematical characterization. We show that two inference modes can coexist within a single model. In-distribution behavior is governed by Bayesian task retrieval, implemented internally through convex combinations of learned task vectors. OOD behavior, by contrast, arises through extrapolative task learning, whose representations occupy a subspace nearly orthogonal to the task-vector subspace. Taken together, our results suggest that task-vector geometry, training distributions, and generalization behaviors are closely related.

## 1 Introduction

Large language models (LLMs) exhibit impressive performance, yet the mechanism underlying their generalization remains unclear. Since the emergence of in-context learning, empirical studies have suggested two modes of task inference [7, 32]: *memorization-based task retrieval*, where the model updates an implicit posterior over in-distribution (ID) training tasks and retrieves previously learned behaviors, and *context-based task generalization*, where it solves out-of-distribution (OOD) tasks whose latents lie outside the training support. However, the internal representations supporting these two modes remain obscure. This motivates two central questions: *How do hidden states summarize context and retrieve memorized tasks? How does the representation geometry change for novel tasks outside the training support?*

To understand these external behaviors, researchers have analyzed internal representations, broadly centering on the *linear representation hypothesis* (LRH) [29, 28, 34], which posits that hidden states encode concepts as linear subspaces. For example, [20] discovered that in ICL, LLMs use in-context examples internally to form task-encoding vectors, commonly called *task vectors*. Throughout, we use "task vector" exclusively in this activation-space sense, as a direction in the hidden-state space, distinct from the weight-space usage in task arithmetic and model merging. Concretely, given a prompt $s = (x_1, f(x_1), x_2, f(x_2), \ldots, x_n, f(x_n), x_{\text{test}})$, an LLM can infer the mapping $f$ from the demonstrations and correctly predict $f(x_{\text{test}})$. Internally, the hidden state in a middle layer contains a task-specific component $\boldsymbol{\theta}_f$ that steers the model toward the correct continuation.

*Code to reproduce all experiments is available at https://github.com/ezyhdxm/mini-ICL.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

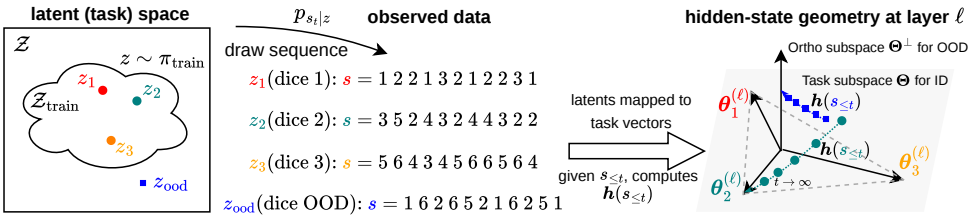



Figure 1: **Connections between data distribution and representation geometry.** This diagram illustrates the synthetic *rolling biased dice* experiment. Each sequence is generated from an unobserved latent $z$ controlling the outcome distribution. A transformer trained on mixture data learns internal task vectors that encode these latents. Two near-orthogonal subspaces largely determine a model's inference mode and generalization behavior.

**Challenges: lack of rigorous foundations for representation geometry.** While prior work on internal representations and task vectors provides valuable insights [18, 6], the mechanistic analysis is often model-dependent and its applicability is rarely verified. For example, various notions of task vectors are proposed [20, 51, 27], sometimes leading to contradictory results in the literature. A central missing piece is a theoretical framework that rigorously defines task vectors and clarifies how they relate to the geometry of internal representations. Without such a foundation, existing results are largely heuristic: task vectors are often introduced operationally or extracted empirically, but their precise meaning, validity, and connection to representation structure remain unclear. As a result, current analyses fail to provide a clear mechanistic picture of how pretrained LLMs internally encode task information, how this encoding is shaped by the training data distribution, and how internal representations mediate the link between external statistical structure and model behavior.

**Bridging the gap via latent-variable modeling.** To develop foundations for task vector representation geometry grounded in rigorous statistical principles, we mainly study synthetic experiments where we train small transformers from scratch on carefully controlled data. As shown in Fig. 1, at a high level, we view tasks as latent variables (or simply, latents) in a space $\mathcal{Z}$. We model training data as a latent-task mixture: for each sequence, we first sample an unobserved latent $z \in \mathcal{Z}_{\text{train}} \subset \mathcal{Z}$ and then sample the sequence from its task-specific distribution $p_z$. Echoing the "multitask learner" heuristic of GPT-2 [39], this latent-variable view is standard in classical text analysis (e.g., topic models) and has recently provided useful insights on ICL [59, 17], induction heads [31, 14], etc.

This setup allows us to broadly define task vectors as mappings of latents, and characterize how a model infers tasks from interactions between hidden states and task vectors. We adopt a "gray-box" perspective that assumes access only to hidden states, making our analysis agnostic to architectural details. Our framework builds on the Markov structure of the data, which yields testable properties of a model's hidden states rather than assumptions (Sec. 4): when these properties hold, task-vector geometry admits a rigorous characterization, and the picture degrades gracefully as the dependence becomes higher-order (Sec. E); when they fail, as in a planted Dyck language whose long-range structure admits no compact summary, common task-vector heuristics break down (Sec. 7). Our approach thus unifies prior work on representation geometry [43, 42] and inference modes [9, 25, 33], and delimits constructively where it applies. Our main findings are summarized below.

1. **A principled framework unifies representation geometry and generalization.** We mathematically define task vectors as representations of latent tasks, and characterize two distinct inference modes: Bayesian task retrieval and extrapolative task learning (Sec. 4).
2. **Task-vector geometry governs Bayesian task retrieval**. For ID sequences, the model infers latent tasks primarily by adjusting the coefficients of task vectors in its hidden states, which eventually converge to the true task vector as sequence length grows (Sec. 5).
3. **Near-orthogonal subspace governs extrapolative task learning.** For OOD sequences, the model can often generalize by using context-related statistics encoded through near-orthogonal subspaces, which emerge only at sufficiently large task diversity (Sec. 6).
4. **The same geometry appears in pretrained LLMs.** Across 12 pretrained models, in-context task representations concentrate in a mid-depth band, and in the persona belief-dynamics setting of Bigelow et al. [4] the simplex coefficients of an instruction-tuned LLM track accumulating in-context evidence like a posterior (Sec. 6.2).

## 2 Related work

**Task vectors in language models.** Task vectors have emerged as a prominent paradigm for explaining how language models extract task information from context and use it to steer subsequent predictions. However, where such vectors should be obtained from, and which internal activations truly constitute task vectors, remains underexplored and still somewhat controversial. Existing work typically extracts task vectors using heuristic procedures such as dummy task prompts [20], averaging across multiple task prompts [51], or opaque optimization-based selection [22], and does so from sources including intermediate ICL hidden states [23], attention-head outputs [60], and MLP activations [27]. However, these construction procedures are largely ad hoc: candidate vectors are typically selected from a small set of layers or components and validated mainly by intervention outcomes, rather than derived from a principled characterization of the underlying data distribution or representation geometry.

**Bayesian inference and OOD generalization.** Early studies observe that LLMs can operate in two modes: retrieve seen tasks and learn new tasks in context [32, 56]. Explanations such as Bayesian beliefs [59, 43] and induction heads [15, 31, 46] focus on either ID or OOD generalization, without reconciling the coexistence of both inference modes. Synthetic experiments in [17, 9, 25, 33] reveal that task diversity in training shapes the inference mode in specific settings, yet no analysis of internal representations is provided. Bayesian latent-variable models, such as PCA [50] and topic models [5], are classical objects in statistics and machine learning, but their generative assumptions have no a priori reason to hold in a transformer, and few works connect them to transformer representations. Closest to our vein are belief-state geometry [43], which regresses hidden states onto the ground-truth belief states of a single hidden Markov process, and the belief-dynamics study of Bigelow et al. [4], which tracks an LLM's belief over latent personas behaviorally as in-context evidence accumulates.

**Near-orthogonal representation geometry.** A central idea in mechanistic interpretability (MI) is the *superposition principle* [16]: models often accommodate a large number of feature vectors by arranging them in near-orthogonal positions, echoing the classical literature in sparse coding [11] and compressed sensing [10, 8]. Near-orthogonal geometry is empirically confirmed in transformers [45, 42] and serves as the foundation for MI work [6, 48, 35, 19]. However, near-orthogonality has not been established as a geometric foundation for the coexistence of ID and OOD inference modes.

## 3 Synthetic experiment setup

**Data generation.** In all synthetic experiments, we first draw a latent $z$ from a finite training set $\mathcal{Z}_{\text{train}} \subset \mathcal{Z}$ and then sample a sequence from the latent-specific distribution.

**E1. Rolling biased dice.** Each latent $z$ corresponds to a probability vector $\boldsymbol{p}$ drawn from a Dirichlet prior $\mathrm{Dir}(\mathbf{1}_6)$. For each sequence, we first sample $\boldsymbol{p}$ and then sample tokens independently from $\boldsymbol{p}$.

**E2. In-context linear regression.** Following [40], we take the latent $z := \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$ as the $D$-dim regression weight vector ($D = 6$). For each sequence, we first sample $\boldsymbol{w}$ and then $T/2$ input-output pairs (treating each vector/scalar as a token) $\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \ldots, \boldsymbol{x}_{T/2}, y_{T/2}$, where $\boldsymbol{x}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D), y_t = \boldsymbol{x}_t^\top \boldsymbol{w} + \varepsilon_t$, and $\varepsilon_t \sim \mathcal{N}(0, 0.25)$ .

**E3. Mixture of Markov chains.** A latent $z := \boldsymbol{P} \in \mathbb{R}^{V \times V}$ ($V = 6$) is a transition matrix for a first-order Markov chain. As in [14], we draw rows of $\boldsymbol{P}$ independently from a Dirichlet prior $\mathrm{Dir}(\mathbf{1}_V)$, then sample the Markov sequence $s_{1:T}$ using $\boldsymbol{P}$, starting from a uniform initial distribution.

**E4. Planted Dyck language.** A Dyck language (Dyck-1) is the set of balanced bracket strings. The latent $z$ is a Dyck string of length $2L$ ($L = 10$, giving $C_{10} = 16{,}796$ latents). Conditioning on $z$, we draw $T$ background tokens i.i.d. uniformly from $\{0, \ldots, V-1\}$ ($V = 6$), select each position independently with probability $\rho = 0.25$, and plant the next character of $z$ there, overwriting the background token; e.g., with $V = 2$ and $z = (()())$, one sequence is $0, 1, (, 0, (, 1, ), 0, (, 1, 1, ), )$.

**Remark 1** (Markov property)**.** *In `E1`–`E3`, sequences are first-order Markov given $z$, so we expect a model to infer the unobserved $z$ by compressing the context into a 'belief' about $z$. The framework extends to finite order, since an order-$w$ chain is first-order on the window of its last $w$ tokens (experiments up to order 10 in Sec. E). `E4`, in contrast, has no finite order: no window summarizes the bracket history, and the properties we study break down there (Sec. 7).*

**Model and training.** We use a decoder-only transformer with rotary embeddings (RoPE) [47]. For experiments E1 and E3, the transformer has 6 layers, 2 heads, and hidden dimension $d = 128$; for E2, we follow [2] by increasing the number of layers to 16 (other hyperparameters remain the same). All models are trained autoregressively with AdamW [24]; see Sec. A for hyperparameters.

# 4 A mathematical framework for representation geometry

This section develops the mathematical framework used throughout the paper. Sec. 4.1 defines task vectors and four geometric properties of hidden states; Sec. 4.2 links these properties to Bayesian task retrieval, while Sec. 4.3 studies extrapolative task learning and shows when it must move beyond the task-vector subspace. Secs. 5–6 provide empirical evidence for the two-mode picture.

## 4.1 Theoretical properties of task-vector geometry

Let $z \in \mathcal{Z}$ be the unobserved latent and $s_1, s_2, \ldots$ be the observed sequence generated from the distribution $P_z$. For a fixed transformer layer $\ell$, denote by $\boldsymbol{h}_t^{(\ell)} \in \mathbb{R}^d$ the residual-stream hidden state at position $t$ after processing the prefix $s_{\le t}$. Throughout this section, we fix a layer and suppress the superscript $\ell$. Since the results below do not depend on the specific parameter space of any experiment, we identify the training latents $z$ with the index set $[K] := \{1, \ldots, K\}$ and write $\boldsymbol{\mu}_t := \mathbb{E}[\boldsymbol{h}_t]$ for the mean hidden state at position $t$. As noted in Sec. 3, tasks E1–E3 satisfy the first-order Markov property: $\mathbb{P}(s_{t+1} \mid z, s_{\le t}) = \mathbb{P}(s_{t+1} \mid z, s_t)$.

Under the Markov property, $\mathbb{P}(s_{t+1} \mid z, s_{\le t})$ depends only on $(z, s_t)$, so a well-trained model's hidden state should likewise be approximately determined by these two factors. Define cell means $\boldsymbol{\mu}_{t,k,a} := \mathbb{E}[\boldsymbol{h}_t \mid z = k, s_t = a]$ and write

$$\boldsymbol{h}_t \;=\; \boldsymbol{\mu}_{t,z,s_t} \;+\; \boldsymbol{\varepsilon}_t, \tag{1}$$

where $\boldsymbol{\varepsilon}_t := \boldsymbol{h}_t - \boldsymbol{\mu}_{t,z,s_t}$ is the residual not explained by $(z, s_t)$. We state four properties characterizing the hidden-state geometry when task retrieval succeeds.

**P0. Long-context stability.** The residual in Eq. 1 vanishes asymptotically: $\mathbb{E}\|\boldsymbol{\varepsilon}_t\|^2 \to 0$ as $t \to \infty$. In other words, the hidden state becomes a deterministic function of $(z, s_t)$, meaning the model has successfully summarized the context. While heuristic task-vector extractions [51, 20] implicitly assume P0, we show in Sec. 7 that this property is not always satisfied.

**P1. Long-context decoupling.** Given P0, we can further ask how the cell means $\boldsymbol{\mu}_{t,k,a}$ are structured. The task-specific component naturally gives rise to the definition of *task vectors* [20].

**Definition 1.** *For a given layer,* task vectors *are* $\{\boldsymbol{\theta}_k\}_{k \le K} \subset \mathbb{R}^d$ *defined by*

$$\boldsymbol{\theta}_k = \lim_{t \to \infty} \big(\mathbb{E}[\boldsymbol{h}_t \mid z = k] - \boldsymbol{\mu}_t\big), \tag{2}$$

*whenever the limit exists. Centering ensures* $\sum_{k \le K} \boldsymbol{\theta}_k = \mathbf{0}$.

P1 requires that the cell means decompose additively into task and token components. There exist token-encoding vectors $\{\boldsymbol{\nu}_a\}_{a \in \mathcal{V}} \subset \mathbb{R}^d$ such that

$$\mathbb{E}\Big\|\boldsymbol{h}_t - \big(\underbrace{\boldsymbol{\mu}_t}_{\text{global mean}} + \underbrace{\boldsymbol{\theta}_z}_{\text{task vector}} + \underbrace{\boldsymbol{\nu}_{s_t}}_{\text{token encoding}}\big)\Big\|_2^2 \to 0 \qquad \text{as } t \to \infty. \tag{3}$$

This additive structure is a form of the *linear representation hypothesis* [34]; in ANOVA terms, P1 requires that the interaction between $z$ and $s_t$ is negligible.

**Remark 2** (Task vectors as class-conditional means)**.** *Definition 1 identifies task vectors with centered class-conditional means of hidden states, the construction used in prior work [20, 51]. Under the additive decomposition of P1, this choice is not special: alternative estimators such as LDA, PCA after conditioning on the current token, and linear probes recover the same vectors (Sec. C.6).*

**P2. Finite-context interpolation.** At finite context, the model represents its knowledge about latent tasks via a convex combination of task vectors. There exist nonnegative coefficients $\{\beta_{t,k}\}_{t \ge 1,\, k \le K}$ satisfying $\sum_{k \le K} \beta_{t,k} = 1$ (depending on context $s_1, \ldots, s_t$) such that

$$\boldsymbol{h}_t \approx \boldsymbol{\mu}_t + \sum_{k \le K} \beta_{t,k}\, \boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t}. \tag{4}$$

P2 formalizes the *superposition principle* [16]: hidden states are linear combinations of task vectors.

**P3. Bayesian posterior alignment.** P3 strengthens P2 by pinning down the same coefficients $\beta_{t,k}$: they approximate the Bayesian posterior,

$$\beta_{t,k} \approx \alpha_{t,k} := \mathbb{P}(z = k \mid s_{\leq t}). \tag{5}$$

This provides a mechanistic characterization of *Bayesian inference*: while the Bayesian perspective has been used to explain ICL [59], its internal geometry has remained unclear. These properties are progressively stronger: P3 $\Rightarrow$ P2, P1 and P1 $\Rightarrow$ P0 in the long-context limit; see Sec. B.1. These properties can be verified empirically. When they hold, they yield task vectors with a precise meaning, recovering the objects described in the literature (Sec. 5); when they fail, as for the planted Dyck language in Sec. 7, they mark the boundary beyond which task vectors do not exist. All empirical claims below are stated for the layer band at which they are evaluated (Sec. 5.1).

### 4.2 Task retrieval: task-vector geometry mirrors Bayesian inference

When a model has memorized a task $z$ during training, it can retrieve the latent task $z$ from test prompts via approximate Bayesian prediction; we call this the **Bayesian task retrieval mode** (M1). Formally, let $\pi_{\text{train}}$ be the prior distribution over $\mathcal{Z}_{\text{train}} \subset \mathcal{Z}$. The optimal prediction under M1 is

$$\mathbb{P}(s_{t+1}{=}a \mid s_{\leq t}){=}\int_{\mathcal{Z}'} \mathbb{P}(s_{t+1}{=}a \mid z, s_t)\, p(z \mid s_{\leq t})\mathrm{d}z,\ \ p(z \mid s_{\leq t}) = \frac{p_z(s_{\leq t})\, \pi_{\text{train}}(z)}{\int_{\mathcal{Z}'} p_z(s_{\leq t})\, \pi_{\text{train}}(z)}, \tag{6}$$

where $\mathcal{Z}' = \mathcal{Z}_{\text{train}}$ and $p(z \mid s_{\leq t})$ is the exact $\alpha_{t,k}$ in Eq. 5. By Eqs. 4 and 5, $\boldsymbol{h}_t$ carries the same posterior weights as Eq. 6: after a shared centering map removes $\boldsymbol{\mu}_t$, the informative component is $\sum_{k\leq K} \alpha_{t,k}(\boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t})$. Thus Bayesian retrieval reduces to a finite lookup-and-mix computation: lookup $\mathbb{P}(\cdot \mid z = k, s_t)$ from the task/token anchors and mix the results with $\alpha_{t,k}$.

**Theorem 1** (Informal; task-vector subspace can implement Bayesian inference)**.** *Suppose that at some layer $\boldsymbol{h}_t$ satisfies P0–P3. Under the structural conditions formalized in Sec. B.2, two additional transformer blocks and a suitable unembedding can make the ID prediction distribution approximate the Bayesian predictive distribution in Eq. 6.*

Sec. B.2 gives the proof. Theorem 1 is about expressivity instead of training dynamics: it guarantees that hidden states satisfying P0–P3 suffice to produce the Bayesian predictive distribution, so the Bayesian interpretation is not vacuous. Sec. 5 confirms empirically that models learn such a solution.

### 4.3 Task learning: OOD inference operates outside task-vector subspace

How can a model infer $z' \notin \mathcal{Z}_{\text{train}}$ from context and achieve OOD generalization? One possibility is that it applies Eq. 6 with an uninformative prior over the full space $\mathcal{Z}$ rather than only the training support, extrapolating beyond seen latents. We call this the **extrapolative task-learning mode** (M2).

M2 relies on context statistics (empirical unigram for E1, ridge solution for E2, and empirical bigram for E3) rather than memorized task vectors (Sec. B.3). The key question is whether such extrapolative inference behavior is enabled by a different geometry compared with M1. We give a theoretical result: when the OOD prediction map has intrinsic dimension $d_0 > k_\star$, where $k_\star$ is the task-vector subspace dimension, a model implementing M2 cannot confine its representations to that subspace.

More formally, suppose $\mathcal{V}$ is discrete and fix $a_\star \in \mathcal{V}$. Let $\boldsymbol{p}(z) := \mathbb{P}(\cdot \mid z, a_\star)$ denote the next-token distribution. Assume P0 holds and that, for $z$ near an interior point $z_0 \in \mathcal{Z}$, the limit $\boldsymbol{\mu}_\star(z) := \lim_{t\to\infty} \boldsymbol{\mu}_{t,z,a_\star}$ exists. Define the task subspace $\mathcal{T} := \operatorname{span}(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K - \boldsymbol{\theta}_1)$, set $k_\star := \dim \mathcal{T}$ and $\mathcal{S} := \boldsymbol{\nu}_{a_\star} + \mathcal{T}$. Let $d_0 := \operatorname{rank} \nabla \boldsymbol{p}(z_0)$ be the local intrinsic dimension of $\boldsymbol{p}$ at $z_0$.

**Theorem 2** (Informal, distance to task subspace lower bound)**.** *Assume P0. Suppose that on some neighborhood $U \ni z_0$, the prediction $\hat{\boldsymbol{p}}(z)$ decoded from $\boldsymbol{\mu}_\star(z)$ is locally $L$-Lipschitz: $\|\hat{\boldsymbol{p}}(z) - \hat{\boldsymbol{p}}(z')\|_2 \leq L\, \|\boldsymbol{\mu}_\star(z) - \boldsymbol{\mu}_\star(z')\|_2$, for all $z, z' \in U$. Additionally, assume that $\|\hat{\boldsymbol{p}}(z) - \boldsymbol{p}(z)\|_2 \leq \delta$, for all $z \in U$ and $\{\boldsymbol{\mu}_\star(z) : z \in U\}$ is bounded. Then there exist local constants $c, C > 0$ such that*

$$\sup_{z\in U} \inf_{\boldsymbol{c}\in\mathcal{S}} \|\boldsymbol{\mu}_\star(z) - \boldsymbol{c}\|_2 \geq \frac{c - \delta}{L} - \frac{C}{2^{d_0/k_\star} - 1}.$$

Table 1: Hidden-state variance decomposition at the final context position. *(a)* Residual variance ratio $\mathrm{SS}_{\mathrm{within}}/\mathrm{SS}_{\mathrm{total}}$: unexplained variance; lower is better. *(b)* Interaction proportion $\eta^2_{\mathrm{interaction}}$: share of $\mathrm{SS}_{\mathrm{between}}$ from task–token interaction; lower supports additivity.

| **Layer** | 0 | 1 | 2 | 3 | 4 | 5 | **Layer** | 4 | 6 | 8 | 10 | 12 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *(a) Residual variance ratio* $\mathrm{SS}_{\mathrm{within}}/\mathrm{SS}_{\mathrm{total}}$ | | | | | | | | | | | | | |
| E1 (Dice) | 0.012 | 0.021 | 0.025 | 0.017 | 0.006 | 0.004 | E2 (Linear) | 0.075 | 0.084 | 0.061 | 0.043 | 0.040 | 0.045 |
| E3 (Markov) | 0.037 | 0.071 | 0.029 | 0.012 | 0.007 | 0.005 | | | | | | | |
| *(b) Interaction proportion* $\eta^2_{\mathrm{interaction}}$ | | | | | | | | | | | | | |
| E1 (Dice) | 0.003 | 0.006 | 0.012 | 0.013 | 0.007 | 0.004 | E2 (Linear) | $<0.001$ | $<0.001$ | 0.007 | 0.040 | 0.081 | 0.117 |
| E3 (Markov) | $<0.001$ | 0.004 | 0.019 | 0.125 | 0.420 | 0.512 | | | | | | | |

The second term in the lower bound decays exponentially in $d_0/k_\star$, so when $d_0 \gg k_\star$ the model cannot implement M2 accurately while staying close to $\mathcal{S}$; smoother decoding (smaller $L$) forces a larger deviation. Note that this result establishes a separation between M2 and the task subspace, but does not force the geometry to be orthogonal by itself. See Sec. B.4 for the formal theorem and Remark 4 therein for more discussion.

## 5 Task-vector geometry represents Bayesian posterior beliefs

Section 4.1 predicts the following behavior under the Bayesian task-retrieval mode (M1).

**Task-vector geometry represents Bayesian inference**: For ID sequences, the model represents training tasks as learned task vectors and updates their mixture weights from context, approximately tracking the Bayesian posterior.

### 5.1 Evaluation results of properties P0–P3

We show that the stated properties of task vectors are mostly satisfied for E1–E3. Fix $K = 3$. For each experiment, we estimate task vectors by replacing the expectation in Eq. 2 with empirical averages $\hat{\mathbb{E}}$ over 1,024 independent sequences per task; estimated quantities use hat notation.

**Evaluating long-context stability.** We assess how much the variance of hidden states is attributed to latent $z \in [K]$ and the current token $a \in \mathcal{V}$. For experiments E1 and E3 with discrete tokens, we calculate $\hat{\boldsymbol{\mu}}_{t,k,a} := \hat{\mathbb{E}}[\boldsymbol{h}_t \mid z = k, s_t = a]$ and $\hat{\boldsymbol{\mu}}_t := \hat{\mathbb{E}}[\boldsymbol{h}_t]$. By a balanced-cell ANOVA diagnostic,

$$\underbrace{\hat{\mathbb{E}}\|\boldsymbol{h}_t - \hat{\boldsymbol{\mu}}_t\|^2}_{\mathrm{SS}_{\mathrm{total}}:\ \widehat{\mathrm{Var}}(\boldsymbol{h}_t)} = \underbrace{(KV)^{-1}\sum_{k,a}\hat{\mathbb{E}}\big[\|\hat{\boldsymbol{\mu}}_{t,k,a} - \hat{\boldsymbol{\mu}}_t\|^2\big]}_{\mathrm{SS}_{\mathrm{between}}} + \underbrace{(KV)^{-1}\sum_{k,a}\hat{\mathbb{E}}\big[\|\boldsymbol{h}_t - \hat{\boldsymbol{\mu}}_{t,k,a}\|^2 | z = k, s_t = a\big]}_{\mathrm{SS}_{\mathrm{within}}:\ \widehat{\mathrm{Var}}(\boldsymbol{h}_t|z,s_t)}.$$

and we measure the residual variance ratio $\mathrm{SS}_{\mathrm{within}}/\mathrm{SS}_{\mathrm{total}}$. This ratio is zero if and only if hidden states $\boldsymbol{h}_t$ are completely determined by latents and current tokens $z, a$. For experiment E2 where tokens are continuously valued, we approximate the proportion of conditional variance using analysis of covariance (ANCOVA) and report the $R^2$ of the linear fit $\boldsymbol{h}_t \approx \boldsymbol{\mu}_t + \boldsymbol{\theta}_k + \boldsymbol{B}_k\boldsymbol{x}_t$ where $\boldsymbol{B}_k$ is a task-specific coefficient matrix. See Sec. C.2 for details.

Table 1(a) confirms that at the last context position the residual variance ratio is small across all layers and all three experiments, so latents and last tokens together account for most of the hidden-state variance at large $t$. Fig. 9 (Sec. C.2) shows the full residual-ratio curves across context positions: the ratio is larger at early positions where limited context hinders task inference, and smaller in later layers where depth aids it.

**Evaluating long-context decoupling.** We assess the additive separability as in Eq. 3 using ANOVA, splitting $\mathrm{SS}_{\mathrm{between}}$ into task, token, and interaction components (Sec. C.2). Table 1(b) reports the interaction proportion $\eta^2_{\mathrm{interaction}}$ at the last context position: it is small across most layers in all three experiments, confirming the additive model and validating P1.

**Evaluating finite-context interpolation.** We test P2 at finite context by fitting the interpolation model in Eq. 4 with simplex-constrained coefficients $\beta_{t,k} \geq 0$, $\sum_k \beta_{t,k} = 1$ (Fig. 2). Across all three experiments, $R^2$ is close to 1 in most layers and positions. Late layers of E2 and E3, however, develop

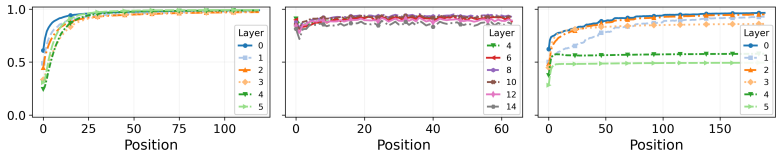



Figure 2: **Finite-context interpolation approximately holds.** $R^2$ of the interpolation model (Eq. 4) across context positions and layers for `E1`, `E2`, `E3`. Values are close to 1 except in late layers and early positions.

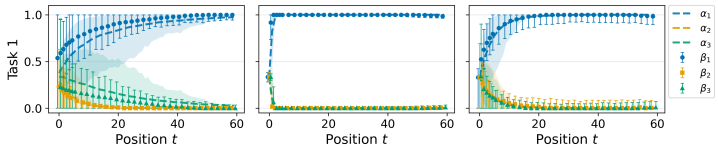



Figure 3: **Bayesian posterior alignment.** Simplex-projected coefficients $\beta_{t,k}$ (markers, 10–90$^{\text{th}}$ percentile error bars) vs. ground-truth posterior $\alpha_{t,k}$ (dashed lines, shaded bands) for `E1`, `E2`, `E3`.

task-specific token encodings (large $\eta^2_{\text{interaction}}$ in Table 1(b) and Fig. 10 in Sec. C.2), causing $R^2$ to degrade, plateauing at 0.60 and 0.52 in the last two layers of `E3`. Sec. C.3 gives a finer-grained variance decomposition by layer. Overall, the interpolation model provides substantial support for `P2`.

**Evaluating Bayesian posterior alignment.** We compare the ground-truth posterior $\alpha_{t,k}$ (Sec. C.4.1) with the model's simplex-projected coefficients $\beta_{t,k}$ (Fig. 3). Across all three tasks, $\beta_{t,k}$ closely tracks $\alpha_{t,k}$ as context accumulates. Sec. C.4.3 confirms that the agreement strengthens in later layers and is not an artifact of the simplex projection.

**Depth dependence.** The empirical claims in this paper refer to a mid-depth band of layers, selected as the layers maximizing held-out ID task-subspace $R^2$, because the properties are not uniform across depth; Remark 5 in Sec. D.12 summarizes how each property varies with depth.

### 5.2 Intervention analysis of task vectors

To go beyond observational analysis, we intervene on hidden representations to test whether task vectors causally control predictions. At each position, we replace the task-subspace component with a target interpolation $\sum_k \alpha^*_{t,k}\,\hat{\boldsymbol{\theta}}_k$, where $\alpha^*_{t,k}$ is randomly drawn from the simplex (see Sec. C.5). This intervention steers the hidden representation to a chosen point in the task-vector simplex. We calculate the difference between steered model outputs and the theoretical predictions based on $\alpha^*_{t,k}$; we also include the unsteered model as a baseline.

Fig. 4 shows that steering moves the model's outputs toward the corresponding mixture predictions: KL divergence drops from 0.11 to 0.03 in `E1` and from 0.23 to 0.06 in `E3`, and RMSE drops from 2.8 to 1.1 in `E2`. This demonstrates that the task-vector subspace provides a controllable representation through which the model interpolates between tasks. Note that the intervention identifies the individual vertices, not merely their span: same-plane bases with a wrong vertex assignment (relabeled or rotated in-plane) fail to steer (KL 0.14–0.20 vs. 0.06; Sec. C.7). See Sec. C.5 for details and baselines.

## 6 Coexistence and geometry of two inference modes

Sec. 5 established that task-vector geometry implements Bayesian retrieval for memorized tasks. We now ask whether extrapolative task learning can coexist with retrieval, and what geometry supports it. Sec. 4.2 and 4.3 hinted that increasing the support of $\mathcal{Z}_{\text{train}}$ shifts the model toward `M2` mode, since the task-vector subspace alone cannot carry task inference over a high-dimensional latent space. To investigate this, we consider the training mixture $\mathcal{Z}_{\text{train}} = \mathcal{Z}_{\text{major}} \cup \mathcal{Z}_{\text{minor}}$: 3 major tasks in $\mathcal{Z}_{\text{major}}$ receive prior 0.9 and $N_{\text{minor}} = 2^m$ minor tasks share 0.1, as described in Sec. D.1. This split underlies the geometric analysis of Sec. 6.1: the major tasks fix a task-vector subspace, and we test whether `M2` operates within it by projecting OOD representations onto this subspace. That geometric question is meaningful only if both modes arise, so we first verify behaviorally that sweeping the diversity exponent $m$ drives the predicted `M1`-to-`M2` transition.

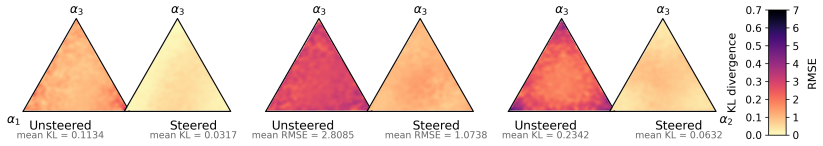



Figure 4: **Causal intervention via the task-vector simplex.** Substituting $\beta_{t,k}$ with randomly drawn $\alpha^*_{t,k}$ steers model outputs to align with the corresponding mixture predictions, confirming causal effects.

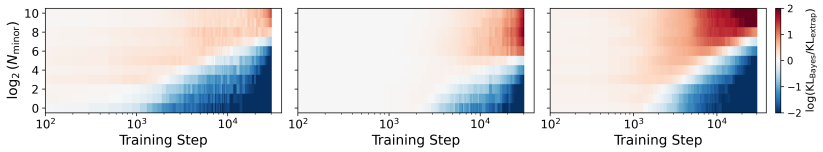



Figure 5: **Phase transition between two inference modes.** Panels (left to right): `E1`, `E2`, `E3`. Blue (negative) indicates that the model is closer to Bayesian task retrieval `M1`, red (positive) indicates closer to the extrapolative task learning `M2`, and white indicates no clear preference. Higher task diversity generally promotes `M2`.

**Task diversity drives the model's inference mode.** We train transformers under varying $m$ and measure each model's predictive distribution against the ideal predictions from both `M1` and `M2` (Eq. 6), yielding two KL divergences under varying $m$ and training steps. Plotting $\log(\mathrm{KL}_{\mathrm{bayesian}}/\mathrm{KL}_{\mathrm{extrapolative}})$ in Fig. 5 reveals clear phase transitions across `E1`–`E3`: early in training and at low diversity the model is better described by `M1`, whereas later in training and at higher diversity it is better described by `M2`. The diagonal boundary indicates that greater task diversity induces the shift earlier in training; red regions also coincide with stronger OOD performance (Sec. D.2). These results indicate that increasing task diversity drives a transition from Bayesian task retrieval toward extrapolative task learning, echoing the results in [40, 25, 33]. Since the minor tasks share a fixed prior mass of $0.1$, the training data per minor task shrinks as $m$ grows, so the transition could reflect either limited data per task or limited model capacity; Sec. D.3 separates the two. Holding the amount of data per minor task fixed as $m$ grows removes the transition over the feasible range, whereas widening the model from $d = 32$ to $512$ or training longer only delays it. The transition is thus driven by data per task rather than model capacity, in line with the diversity threshold of Raventós et al. [40] and the transient generalizing solutions of Carroll et al. [9].

### 6.1 Coexisting near-orthogonal subspaces

We hypothesize a stronger geometry for the `M2` inference mode than Theorem 2 suggests.

> **Near-orthogonal representation hypothesis**: a transformer internally encodes both inference modes via subspaces, with Bayesian task retrieval carried by the task-vector subspace and extrapolative task learning by a nearly orthogonal subspace.

To test this hypothesis, we train transformers for varying $m$ and extract task vectors as in Sec. 5. For averaged hidden states from OOD sequences, we compute the projection $R^2$ onto the major task subspace $\mathrm{col}(\hat{\mathbf{\Theta}})$: $R^2 = 0$ indicates orthogonality, $R^2 = 1$ full alignment.

**Results.** Fig. 6 reports $R^2$ over training for varying $m$. At low diversity $R^2$ remains high: the major task subspace still explains part of the computation on OOD sequences. As $m$ grows, $R^2$ drops consistently across `E1`–`E3`, and at large diversity OOD representations are nearly orthogonal to $\mathrm{col}(\hat{\mathbf{\Theta}})$. This separation is *learned*: all curves start at high $R^2$, and the drop emerges progressively only for large $m$ (additional plots in Sec. D.4). Since a random subspace in high dimension is already nearly orthogonal to any fixed low-dimensional one, a low $R^2$ alone is weak evidence; Sec. D.11 therefore compares against a random-subspace null and shows that the OOD-relevant subspace starts inside the task subspace, moves to chance-level overlap as `M2` develops, and, unlike a random subspace, is necessary for OOD prediction. Comparing Figs. 5 and 6 reveals the same two-phase dynamic across `E1`–`E3`: an undifferentiated phase (no KL preference between `M1`/`M2`, uniformly high $R^2$) followed by diversity-driven specialization, in which high-$m$ models move toward `M2` and $R^2$ drops toward zero; the ID and OOD losses show the same pattern (Sec. D.2). This staged pattern echoes the two-phase dynamics of grokking [54] and attention-head differentiation [55].

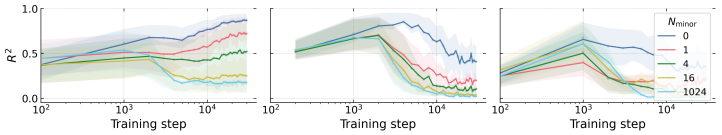



Figure 6: **Projection $R^2$ of OOD hidden states on major task subspace.** Panels: E1, E2, E3. As task diversity increases, projection $R^2$ becomes smaller, supporting the near-orthogonal representation hypothesis.

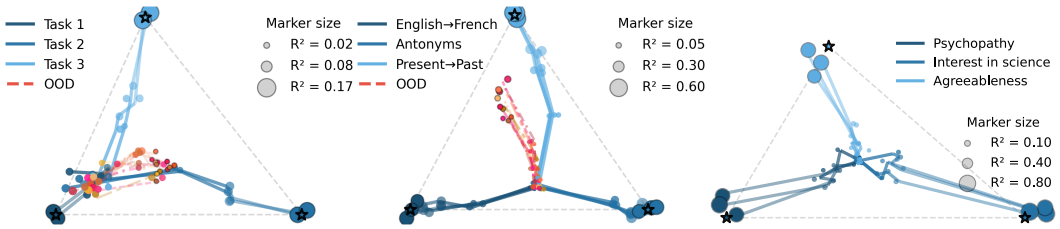



Figure 7: **Task-subspace trajectories for E3 (left), Qwen2.5-7B word-function tasks (middle), and Llama-3.1-8B-Instruct personas (right).** Task vectors of 3 tasks form a triangle onto which hidden states are projected; marker size encodes the projection $R^2$. ID prompts: $\boldsymbol{h}_t$ converges toward the true task vertex as context grows. OOD prompts (middle): hidden states remain near-orthogonal to the task subspace (small $R^2$). Personas (right): contexts of $N = 0$–$128$ persona statements move from the center to the persona's vertex, reaching $R^2 \approx 0.85$.

### 6.2 Pretrained language models

**Word-function tasks in pretrained LLMs.** We ask whether the same geometry appears in pretrained models, using the 12 word-function tasks of Todd et al. [51] (e.g., English→French, Antonyms) as ID tasks and, as OOD tasks, variants that apply an unnatural transform to the input word or to the answer. We ran this analysis for all 12 tasks, every layer, and 12 pretrained models from 7 families (0.5B–8B); Fig. 7 (middle) shows one setting, Qwen2.5-7B at layer 20 with three ID tasks and their OOD variants, next to the E3 trajectories (left). In both, ID prompts converge toward the true task vertex as demonstrations accumulate, while OOD prompts stay nearly orthogonal to the task subspace (small $R^2$). All 12 models show the same pattern, with the ID task-subspace $R^2$ peaking at mid-depth; Gemma-3-4B is the exception (Secs. D.5–D.6).

**Persona belief dynamics.** Bigelow et al. [4] study how an instruction-tuned LLM's belief about the persona it is speaking as evolves with in-context evidence. In our framework, the persona is the latent task, the statements it would make are samples from that task, and the belief over personas is the posterior over tasks. Using their persona data [36], a prompt lists $N$ statements of one persona, followed by the cue "Another statement about myself:", and we take the hidden state of Llama-3.1-8B-Instruct at the cue (layer 6; details in Sec. D.7). Task vectors for $K = 3$ personas are built from held-out statements, and each hidden state is fit as a sum-to-one combination of them, whose coefficients we read as the model's belief. As $N$ grows from 0 to 128, the coefficient on the context persona rises from $1/3$ to $0.95$ while the others fall to near zero, and the hidden states move from the center of the simplex to the persona's vertex (Fig. 7, right). Lacking an oracle posterior, this is qualitative rather than formal evidence for P3 (replicated on Qwen2.5-7B-Instruct in Sec. D.7).

### 6.3 Causal interventions on subspaces

To validate the causal role of the two subspaces, we intervene on hidden states and measure task-specific losses. Let $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}}$ denote the fractional loss degradation for $\text{mode} \in \{\text{maj}, \text{ood}\}$, where $g_{\text{mode}}$ is the total in-context loss reduction, so that $100\%$ means all ICL for that mode is erased.

**Task-subspace intervention.** With $\boldsymbol{P}_{\text{task}}$ projecting onto $\text{col}(\hat{\boldsymbol{\Theta}})$, $\boldsymbol{h} \leftarrow \boldsymbol{h} - \gamma \boldsymbol{P}_{\text{task}} \boldsymbol{h}$ selectively degrades major-task performance while largely preserving OOD losses (Table 2, left), showing that Bayesian task retrieval depends on $\text{col}(\hat{\boldsymbol{\Theta}})$.

**Orthogonal subspace intervention.** Within $\text{col}(\hat{\boldsymbol{\Theta}})^{\perp}$ we identify a low-rank subspace $\hat{\boldsymbol{V}}_{\text{opt}}$ via optimization on held-out minor-task data and apply $\boldsymbol{h} \leftarrow \boldsymbol{h} - \gamma \hat{\boldsymbol{V}}_{\text{opt}} \hat{\boldsymbol{V}}_{\text{opt}}^{\top} \boldsymbol{h}$ ($\gamma > 1$). Table 2 (right) shows large, selective loss increases for OOD tasks, approaching $100\%$ of the OOD in-context gain, while major-task loss and a random same-rank baseline remain near zero. This double dissociation establishes a near-orthogonal mechanism for extrapolative task learning. It holds across the mid-depth

Table 2: **Causal intervention on subspaces confirms near-orthogonal representations.** Relative loss degradation $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}} \times 100\%$, averaged over the mid-layer band listed per experiment (criterion in Sec. 5.1); per-layer curves in Sec. D.8, frozen-band replication in Sec. D.12. "Rand." is the mean over 5 random same-rank directions from $\text{col}(\hat{\boldsymbol{\Theta}})^{\perp}$.

| | | Task Subspace Interv. (%) | | Orth. Subspace Interv. (%) | | |
|---|---|---|---|---|---|---|
| Exp. | Layers | Maj. | OOD | Maj. | OOD | Rand. |
| E1 | 1–4 | 211.1 | 12.1 | 4.0 | 95.8 | 2.9 |
| E3 | 2–4 | 132.0 | 13.1 | 2.1 | 99.6 | 3.0 |
| E2 | 9–12 | 83.6 | −1.0 | 3.0 | 85.4 | 1.9 |

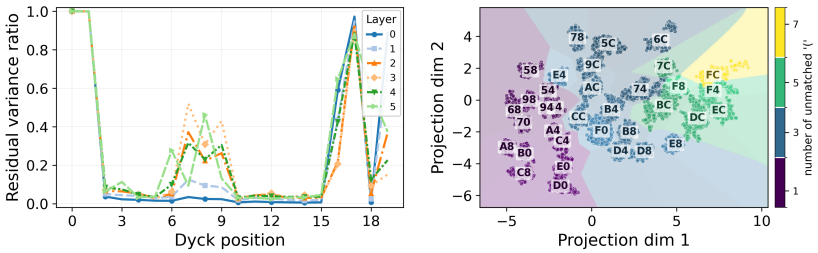



Figure 8: **Planted Dyck language (E4) exhibits long-context dependence and prefix memorization.** ***Left***: residual variance ratio of $\boldsymbol{h}_t$ after conditioning on $z$ and the last 3 planted characters at each Dyck position. Unlike E1–E3, a local window leaves substantial residual variance, spiking where long-range bracket matching is required. ***Right***: 2D projection of final-layer hidden states at Dyck prefix length $l = 7$ (35 prefix classes, labeled by hexadecimal codes; see Fig. 35).

range and is lost only at the last layers, where the shared prediction signal dominates (Secs. D.8 and D.11). Sec. D.9 gives details; Sec. D.10 shows that $\hat{\boldsymbol{V}}_{\text{opt}}^{\top}\boldsymbol{h}$ linearly encodes context statistics.

## 7 Non-Markovian case: when hidden states don't summarize context

The previous sections rely on a Markovian premise. Here we present a non-Markovian counterexample to show why that premise matters. When long-range structure breaks local summarization, even the commonly assumed property P0 fails. Such failure requires dependence of unbounded order: for Markov chains of finite order $w$, the M1–M2 transition, P0–P3 (with a window of $w$ tokens), and the near-orthogonal dissociation persist up to order 10 (Sec. E), whereas the Dyck language, which has no finite order, lies beyond this range. This demonstrates that the validity of common heuristics depends sensitively on the data-generating assumptions, highlighting the need for a principled framework and the value of its theoretical grounding.

**Long-context dependence.** We train a small autoregressive transformer to near-optimal loss and, on held-out sequences, measure the variance of $\boldsymbol{h}_t$ conditioned on the latent $z$ and a local window of preceding tokens. In Fig. 8 (left), the residual variance ratio (hidden-state variance *not* explained by $z$ and the last 3 planted characters) at each Dyck position $j$ (the $j$-th planted bracket) spikes to nearly 1 where long-range bracket matching is required and drops to near zero elsewhere. The 2D projection in Fig. 8 (right) further shows that the model encodes the full Dyck prefix at each position rather than a simple summary such as the running open-minus-close count. Conditioning on that count leaves 58–84% of the variance unexplained, versus about 3% for the full prefix (Sec. E). See App. F for the prefix-probe construction and validation accuracies.

## 8 Limitations and Future Work

This work presents a modest attempt to bridge generalization and representation geometry in transformers. First, our analysis of E1–E3 relies on the Markov property of synthetic data; which natural data admit a compact task statistic remains open. Second, our near-orthogonal subspace does not provide a mechanistic account of OOD generalization, which is complex in general, and its orthogonality lacks a theoretical explanation. Third, our empirical claims are stated for a mid-depth band of layers, and how the task representation is re-encoded at the last layers is left open. Fourth, our pretrained-LLM evidence is observational, and causal validation in pretrained models, including steering along persona task vectors, remains to be done. Finally, our analysis is theory-inspired, but not yet a formal theory of generalization or learning dynamics, left to future work.

## Acknowledgments and Disclosure of Funding

We thank Keith Levin, Karl Rohe, Joshua Cape, Hanbaek Lyu, Sebastien Roch, Robert Nowak, and Liang Twist Shan for helpful discussions, and the anonymous reviewers and area chair for feedback that improved the paper. Y.Z. is partially supported by NSF-DMS grant 2412052 and by a Coefficient Giving (formerly Open Philanthropy) grant. The authors declare no competing interests.

## References

[1] J. Aitchison. *The Statistical Analysis of Compositional Data*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1986.

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

[3] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[4] Eric Bigelow, Daniel Wurgaft, YingQiao Wang, Noah Goodman, Tomer Ullman, Hidenori Tanaka, and Ekdeep Singh Lubana. Belief dynamics reveal the dual nature of in-context learning and activation steering. *arXiv preprint arXiv:2511.00617*, 2025.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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

[8] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[9] Liam Carroll, Jesse Hoogland, Matthew Farrugia-Roberts, and Daniel Murfet. Dynamics of transient structure in in-context linear regression transformers. *arXiv preprint arXiv:2501.17745*, 2025.

[10] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4): 1289–1306, 2006.

[11] David L Donoho and Philip B Stark. Uncertainty principles and signal recovery. *SIAM Journal on Applied Mathematics*, 49(3):906–931, 1989.

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[13] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the $\ell_1$-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 272–279, 2008. doi: 10.1145/1390156.1390191.

[14] Ezra Edelman, Nikolaos Tsilivis, Benjamin L Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. *Advances in neural information processing systems*, 37:64273–64311, 2024.

[15] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[16] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

[17] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

[18] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021.

[19] Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. When models manipulate manifolds: The geometry of a counting task. *arXiv preprint arXiv:2601.04480*, 2026.

[20] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

[21] Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of LLMs: Stages of inference? *arXiv preprint arXiv:2406.19384*, 2024.

[22] Dongfang Li, Zhenyu Liu, Xinshuo Hu, Zetian Sun, Baotian Hu, and Min Zhang. In-context learning state vector with inner and momentum optimization. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0ed52d7f6f641f228405d48a611e0684-Abstract-Conference.html.

[23] 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*, pages 32287–32307. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bx.html.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[25] Yue M Lu, Mary Letey, Jacob A Zavatone-Veth, Anindita Maiti, and Cengiz Pehlevan. Asymptotic theory of in-context learning by linear attention. *Proceedings of the National Academy of Sciences*, 122(28):e2502599122, 2025.

[26] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[27] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5030–5047. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.281. URL https://aclanthology.org/2024.naacl-long.281/.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[29] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1090/.

[30] nostalgebraist. Interpreting GPT: the logit lens. LessWrong, https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020.

[31] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[32] Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, 2023.

[33] Core Francisco Park, Ekdeep Singh Lubana, and Hidenori Tanaka. Competition dynamics shape algorithmic phases of in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XgH1wfHSX8.

[34] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning*, 2024.

[35] Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bVTM2QKYuA.

[36] Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, 2023.

[37] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8: 143–195, 1999.

[38] Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

[41] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt 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)*, pages 15504–15522, 2024.

[42] Adam Shai, Loren Amdahl-Culleton, Casper L Christensen, Henry R Bigelow, Fernando E Rosas, Alexander B Boyd, Eric A Alt, Kyle J Ray, and Paul M Riechers. Transformers learn factored representations. *arXiv preprint arXiv:2602.02385*, 2026.

[43] Adam S Shai, Sarah E Marzen, Lucas Teixeira, Alexander G Oldenziel, and Paul M Riechers. Transformers represent belief state geometry in their residual stream. *Advances in Neural Information Processing Systems*, 37:75012–75034, 2024.

[44] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning*, 2025.

[45] Jiajun Song and Yiqiao Zhong. Uncovering hidden geometry in transformers via disentangling position and context. *arXiv preprint arXiv:2310.04861*, 2023.

[46] Jiajun Song, Zhuoyan Xu, and Yiqiao Zhong. Out-of-distribution generalization via composition: A lens through induction heads in transformers. *Proceedings of the National Academy of Sciences*, 122(6):e2417182122, 2025. doi: 10.1073/pnas.2417182122. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2417182122`.

[47] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[48] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

[49] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, 2019.

[50] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

[51] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[52] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[53] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[54] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad217e0c7fecc71bdf48660ad6714b07-Abstract-Conference.html`.

[55] George Wang, Jesse Hoogland, Stan van Wingerden, Zach Furman, and Daniel Murfet. Differentiation and specialization of attention heads via the refined local learning coefficient. In *International Conference on Learning Representations*, 2025.

[56] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[57] Lloyd Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

[58] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

[59] 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*, 2022.

[60] Kayo Yin and Jacob Steinhardt. Which attention heads matter for in-context learning? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 72428–72461. PMLR, 2025. URL https://proceedings.mlr.press/v267/yin25e.html.

**Organization of the appendix.** We organize our appendix into five sections, following the sequential order of the main text. Specifically,

- Sec. A provides the architecture and training hyperparameters used throughout our synthetic experiments (complementing the Setup in Sec. 3).
- Sec. B contains the theoretical material supporting Sec. 4: the chain of implications among Properties P0–P3 (Sec. B.1), the formal proof of Theorem 1 on Bayesian realizability (Sec. B.2), explicit formulas for the extrapolative mode M2 (Sec. B.3), and the proof of Theorem 2 on the simplex packing obstruction (Sec. B.4).
- Sec. C provides the computational and empirical details behind the evaluation of Properties P0–P3 and the simplex intervention reported in Sec. 5, together with alternative task-vector estimators (Sec. C.6) and the vertex-identity controls for the intervention (Sec. C.7).
- Sec. D collects additional evidence and details for the two-mode picture in Sec. 6: the major/minor training mixture, ID/OOD loss dynamics, the capacity versus per-task-data sweeps (Sec. D.3), simplex trajectories of hidden states, the pretrained-LLM experiments (Qwen2.5-7B, the twelve-model sweep in Sec. D.6, and the persona belief-dynamics study in Sec. D.7), per-layer breakdowns of the causal interventions, the principal-angle analysis showing that the near-orthogonality of the OOD subspace is learned (Sec. D.11), and the layer-band criterion with a replication of Table 2 using fixed bands (Sec. D.12).
- Sec. E extends the analysis beyond first-order Markov data: finite-order sweeps of the transition, P0–P3, and the interventions, and the conditional-variance measurement for Dyck.
- Sec. F reports the prefix-probe analysis of the planted Dyck experiment (E4) discussed in Sec. 7.

# A Experiment Details

Code for all experiments, including data generation, training, and the analyses reported in this paper, is available at https://github.com/ezyhdxm/mini-ICL.

**Architecture and training hyperparameters.** All four experiments use decoder-only transformers with pre-layer-norm, rotary positional encodings, and a $4\times$ MLP expansion ratio, trained autoregressively with AdamW. The biased dice (E1), mixture of Markov chains (E3), and planted Dyck (E4) share a 6-layer, 128-dimensional backbone, while the in-context linear regression (E2) uses a deeper

16-layer model. Per-experiment values for depth, width, context length, optimizer, learning-rate schedule, and other training settings are listed in Tables 3 and 4; the resulting checkpoints are the ones reused in every subsequent empirical analysis.

**Compute and runtime.** All training and analyses are run on a single NVIDIA RTX 3090 (24 GB). A full 30 k–32 k-step training run takes roughly 10 minutes for `E1`, 13 minutes for `E3` and `E4`, and 50 minutes for `E2` (whose 16-layer backbone dominates the compute). The two-mode and orthogonal-subspace analyses (Sec. 6 and Sec. D) sweep the minor-pool size $N_{\text{minor}} \in \{0, 1, 2, 4, \ldots, 1024\}$ and retrain the model for each value, so a single-seed sweep takes about 2 GPU-hours for `E1`, 2.5 for `E3`, and 9 for `E2`. Downstream analyses (task-vector extraction, projection $R^2$, simplex and orthogonal interventions, posterior alignment) all complete in well under one GPU-hour per checkpoint.

Table 3: Architecture hyperparameters. All models are decoder-only transformers with pre-layer-norm and $4\times$ MLP expansion.

| | Layers | Heads | $d$ | FF dim | $T$ | Pos. enc. | Batch |
|---|---|---|---|---|---|---|---|
| Biased dice | 6 | 2 | 128 | 512 | 128 | rotary | 128 |
| Markov chains | 6 | 2 | 128 | 512 | 192 | rotary | 128 |
| Planted Dyck | 6 | 2 | 128 | 512 | 192 | rotary | 128 |
| Linear regression | 16 | 2 | 128 | 512 | 128 | rotary | 256 |

Table 4: Training hyperparameters. All experiments use AdamW. The triangle schedule linearly increases the learning rate during warmup and then linearly decays it to the minimum; the cosine schedule uses cosine annealing after warmup.

| | Steps | LR | Weight decay | Warmup | Schedule | Grad clip | Min LR |
|---|---|---|---|---|---|---|---|
| Biased dice | 30 k | $4\times10^{-4}$ | $4\times10^{-4}$ | 15 k | triangle | — | $10^{-5}$ |
| Markov chains | 30 k | $4\times10^{-4}$ | $4\times10^{-4}$ | 15 k | triangle | — | $10^{-5}$ |
| Planted Dyck | 32 k | $2\times10^{-4}$ | $2\times10^{-4}$ | 8 k | cosine | 0.4 | $2\times10^{-5}$ |
| Linear regression | 30 k | $2\times10^{-4}$ | $1\times10^{-4}$ | 15 k | triangle | — | $10^{-5}$ |

# B Theoretical Foundations of the Property Framework

This section collects the proofs and additional derivations that support the mathematical framework introduced in Sec. 4.

- Sec. B.1 establishes the chain of implications among Properties `P0`–`P3`.
- Sec. B.2 provides the formal statement and proof of Theorem 1 on Bayesian realizability.
- Sec. B.3 derives the closed-form predictors that underlie the extrapolative task-learning mode `M2`.
- Sec. B.4 proves the simplex packing obstruction (Theorem 2).

## B.1 Theoretical relationship between properties `P0`–`P3`

This subsection establishes the implication chain among the four geometric properties `P0`–`P3` introduced in Sec. 4.1:

$$\texttt{P3} \;\Rightarrow\; \texttt{P2}, \qquad \texttt{P3} \;\Rightarrow\; \texttt{P1} \text{ in the long-context limit}, \qquad \texttt{P1} \;\Rightarrow\; \texttt{P0}.$$

This justifies treating the empirical evidence for `P3` reported in Sec. 5.1 as evidence for the weaker properties `P0`–`P2` as well.

**Proposition 1** (`P3` $\Rightarrow$ `P2`)**.** *If `P3` holds, then `P2` holds.*

*Proof.* By `P3`, the hidden state admits the decomposition $\boldsymbol{h}_t \approx \boldsymbol{\mu}_t + \sum_{k\le K} \beta_{t,k}\,\boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t}$ with coefficients $\beta_{t,k} \approx \alpha_{t,k} := \mathbb{P}(z = k \mid s_{\le t})$. Because $\{\alpha_{t,k}\}_{k\le K}$ is a probability distribution, we have $\alpha_{t,k} \ge 0$ and $\sum_{k\le K} \alpha_{t,k} = 1$, so the coefficients $\beta_{t,k}$ are nonnegative and sum to one. This is precisely the simplex constraint required by `P2`. □

**Proposition 2** (P3 $\Rightarrow$ P1 in the long-context limit)**.** *Suppose P3 holds and the posterior concentrates on the true latent, i.e., $\alpha_{t,k} \to \delta_{k,z}$ almost surely as $t \to \infty$, where $z$ denotes the true latent. Then P1 holds in the limit $t \to \infty$.*

*Proof.* Substituting $\alpha_{t,k} \to \delta_{k,z}$ into the P3 decomposition yields

$$\boldsymbol{h}_t \;\approx\; \boldsymbol{\mu}_t + \sum_{k \le K} \alpha_{t,k}\, \boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t} \;\longrightarrow\; \boldsymbol{\mu}_t + \boldsymbol{\theta}_z + \boldsymbol{\nu}_{s_t},$$

recovering exactly the additive structure required by P1. The hypothesis $\alpha_{t,k} \to \delta_{k,z}$ a.s. holds in our experiments by the ergodic theorem for finite-state Markov chains (E1, E3) and an analogous argument for E2 [e.g., 52, Ch. 10]. □

**Proposition 3** (P1 $\Rightarrow$ P0)**.** *If P1 holds, then P0 holds.*

*Proof.* Write $\boldsymbol{\varepsilon}_t = \boldsymbol{h}_t - \boldsymbol{\mu}_{t,z,s_t}$, where $\boldsymbol{\mu}_{t,z,s_t} = \mathbb{E}[\boldsymbol{h}_t \mid z, s_t]$ is the cell mean. Because the conditional expectation minimizes mean squared error over all functions of $(z, s_t)$, we have

$$\mathbb{E}\big\|\boldsymbol{\varepsilon}_t\big\|^2 = \mathbb{E}\big\|\boldsymbol{h}_t - \boldsymbol{\mu}_{t,z,s_t}\big\|^2 \;\le\; \mathbb{E}\big\|\boldsymbol{h}_t - g(z, s_t)\big\|^2$$

for every measurable $g\colon [K] \times \mathcal{V} \to \mathbb{R}^d$. Taking $g(z, s_t) = \boldsymbol{\mu}_t + \boldsymbol{\theta}_z + \boldsymbol{\nu}_{s_t}$, which depends on $(z, s_t)$ alone, gives

$$\mathbb{E}\big\|\boldsymbol{\varepsilon}_t\big\|^2 \;\le\; \mathbb{E}\big\|\boldsymbol{h}_t - (\boldsymbol{\mu}_t + \boldsymbol{\theta}_z + \boldsymbol{\nu}_{s_t})\big\|^2 \;\longrightarrow\; 0 \qquad \text{as } t \to \infty,$$

where the right-hand side vanishes by the P1 guarantee (Eq. 3). □

### B.2 Proof of Theorem 1 in Sec. 4.2

This subsection gives the formal statement and proof of Theorem 1 from Sec. 4.2. It shows that posterior-aligned representations (properties P2–P3) are sufficient for a transformer to approximate the Bayesian predictive distribution under the task-retrieval mode M1. The high-level idea is that, once the residual stream encodes the posterior weights $\alpha_{t,k}$ as the coefficients in front of the task vectors $\boldsymbol{\theta}_k$, the remaining Bayesian computation

$$\boldsymbol{q}_{t+1} \;=\; \sum_{k=1}^{K} \alpha_{t,k}\, \boldsymbol{p}_{k,s_t}$$

reduces to two operations, each realizable by one transformer block: (i) a finite lookup that mixes the per-pair conditionals $\boldsymbol{p}_{k,a}$ into the residual stream, followed by (ii) an approximate coordinatewise logarithm whose softmax recovers $\boldsymbol{q}_{t+1}$. The output unembedding is also part of the construction below, so the theorem does not assume any property of a pre-existing unembedding matrix. The full construction is given in Sec. B.2.1 and relies on two technical lemmas stated and proved in Sec. B.2.2.

#### B.2.1 Detailed Statement of Theorem 1 and Proof

We now state the formal version of Theorem 1, an *expressivity* statement. Posterior-aligned residuals (P2–P3) together with the structural conditions (i)–(iv) below suffice for two appended transformer blocks to approximate the Bayesian predictive $\boldsymbol{q}_{t+1}$ within $\ell_\infty$ tolerance $\delta$. The lookup MLP in Block 1 has hidden width $KV$. Block 2 has hidden width $m_{\log}(\delta, \varepsilon, V) + 2V$, where $m_{\log}$ is the width of a coordinatewise ReLU spline approximation to $\log$ on $[\varepsilon, 1]$ at the accuracy used below. These are constructive upper bounds; we do not claim width optimality, nor do we address whether training finds this configuration or whether conditions (i)–(iv) hold for arbitrary trained models.

**Theorem 3** (Internal geometry mirrors Bayesian inference; formal version of Theorem 1)**.** *Consider a LayerNorm-free transformer with residual-stream dimension $d$, vocabulary $\mathcal{V}$ of size $V$ (identified with $[V]$), and training latent set $\mathcal{Z}_{\text{train}} = [K]$. Suppose the following conditions hold.*

*(i)* ***Uniform posterior-aligned representation.*** *At some layer $\ell$, fix a set of positions $\mathcal{I}$. For every $t \in \mathcal{I}$, the residual stream satisfies*

$$\boldsymbol{h}_t = \boldsymbol{\mu}_t + \sum_{k=1}^{K} \alpha_{t,k} \boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t}, \qquad \alpha_{t,k} = \mathbb{P}(z = k \mid s_{\le t}),$$

*where $\{\boldsymbol{\theta}_k\}_{k=1}^K \subset \mathbb{R}^d$ are task vectors and $\{\boldsymbol{\nu}_a\}_{a\in\mathcal{V}} \subset \mathbb{R}^d$ are token-encoding vectors (cf. properties P2–P3 in Sec. 4.1).*

*(ii)* ***Shared centering map.*** *Let*

$$\mathcal{M}_{\mathcal{I}} := \operatorname{span}\{\boldsymbol{\mu}_t : t \in \mathcal{I}\}, \qquad \mathcal{U} := \operatorname{span}\big(\{\boldsymbol{\theta}_k\}_{k\le K} \cup \{\boldsymbol{\nu}_a\}_{a\in\mathcal{V}}\big).$$

*Assume $\mathcal{M}_{\mathcal{I}} \cap \mathcal{U} = \{0\}$. Equivalently, there exists a fixed linear map $\boldsymbol{C} : \mathbb{R}^d \to \mathbb{R}^d$ such that*

$$\boldsymbol{C}\boldsymbol{\mu}_t = 0, \qquad \boldsymbol{C}\boldsymbol{\theta}_k = \boldsymbol{\theta}_k, \qquad \boldsymbol{C}\boldsymbol{\nu}_a = \boldsymbol{\nu}_a$$

*for all $t \in \mathcal{I}$, $k \le K$, and $a \in \mathcal{V}$. We call such a map a shared centering map.*

*(iii)* ***Linear independence.*** *The family*

$$\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\} \cup \{\boldsymbol{\nu}_a : a \in \mathcal{V}\}$$

*is linearly independent in $\mathbb{R}^d$; in particular $d \ge K + V$.*

*(iv)* ***Bounded-away transition probabilities.*** *There exists $\varepsilon \in (0, 1/V)$ such that*

$$\mathbb{P}(s_{t+1} = b \mid z = k,\, s_t = a) \ge \varepsilon$$

*for all $t \in \mathcal{I}$, $b \in \mathcal{V}$, $k \in [K]$, and $a \in \mathcal{V}$.*

*Then for every $\delta > 0$, there exist an output unembedding $\boldsymbol{W}_{\mathrm{U}} \in \mathbb{R}^{V\times d}$ and two additional transformer blocks, all shared across $t \in \mathcal{I}$, with the blocks appended after layer $\ell$, such that for every $t \in \mathcal{I}$ the model's next-token prediction $\hat{\boldsymbol{p}}_{t+1} \in \mathbb{R}^V$ satisfies*

$$\|\hat{\boldsymbol{p}}_{t+1} - \boldsymbol{q}_{t+1}\|_\infty < \delta,$$

*where*

$$\boldsymbol{q}_{t+1} := \sum_{k=1}^K \alpha_{t,k} \boldsymbol{p}_{k,s_t}, \qquad \boldsymbol{p}_{k,a} := \big(\mathbb{P}(s_{t+1} = b \mid z = k,\, s_t = a)\big)_{b\in\mathcal{V}}.$$

*Moreover, the first appended MLP has hidden width $KV$, and the second has hidden width $m_{\log}(\delta, \varepsilon, V) + 2V$.*

*Proof.* We first fix a gauge for the construction. Although task and token effects are defined earlier in a centered gauge, the lookup construction is most convenient in an uncentered gauge. In particular, we do not impose $\sum_{k=1}^K \boldsymbol{\theta}_k = 0$ or $\sum_{a\in\mathcal{V}} \boldsymbol{\nu}_a = 0$, since either constraint would create a nontrivial linear dependence and conflict with condition (iii). All structural assumptions in the theorem, including the shared-centering map $\boldsymbol{C}$, are understood in this uncentered gauge. This is only a change of coordinates at the level of represented states: setting

$$\bar{\boldsymbol{\theta}} := K^{-1} \sum_{k=1}^K \boldsymbol{\theta}_k, \qquad \bar{\boldsymbol{\nu}} := V^{-1} \sum_{a\in\mathcal{V}} \boldsymbol{\nu}_a,$$

and replacing

$$(\boldsymbol{\mu}_t, \boldsymbol{\theta}_k, \boldsymbol{\nu}_a) \mapsto (\boldsymbol{\mu}_t + \bar{\boldsymbol{\theta}} + \bar{\boldsymbol{\nu}},\, \boldsymbol{\theta}_k - \bar{\boldsymbol{\theta}},\, \boldsymbol{\nu}_a - \bar{\boldsymbol{\nu}})$$

leaves $\boldsymbol{\mu}_t + \sum_k \alpha_k \boldsymbol{\theta}_k + \boldsymbol{\nu}_a$ unchanged for every $\boldsymbol{\alpha} \in \Delta^{K-1}$ and token $a$.

We append two transformer blocks after layer $\ell$. In both blocks, set the attention output projection to zero, so each attention sublayer contributes only the residual identity. It remains to construct the two MLP sublayers and the output unembedding.

Let $\boldsymbol{x}_t := \boldsymbol{h}_t$ be the input to the first appended block and set

$$\boldsymbol{u}_t := \boldsymbol{C}\boldsymbol{x}_t.$$

By conditions (i)–(ii),

$$\boldsymbol{u}_t = \sum_{k=1}^K \alpha_{t,k} \boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t} = \sum_{k=1}^K \alpha_{t,k} (\boldsymbol{\theta}_k + \boldsymbol{\nu}_{s_t}).$$

Write

$$\boldsymbol{u}_{k,a} := \boldsymbol{\theta}_k + \boldsymbol{\nu}_a.$$

Also define the token-embedding matrix

$$\boldsymbol{E} := (\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_V) \in \mathbb{R}^{d\times V}.$$

By condition (iii), $\boldsymbol{E}$ has full column rank.

**Step 1: Build the Bayesian mixture while keeping $\boldsymbol{\mu}_t$.** Define

$$\boldsymbol{f}(k,a) := \boldsymbol{E}\boldsymbol{p}_{k,a} - \boldsymbol{u}_{k,a} = \boldsymbol{E}\boldsymbol{p}_{k,a} - \boldsymbol{\theta}_k - \boldsymbol{\nu}_a.$$

By Lemma 1, there is a width-$KV$ two-layer ReLU MLP $\texttt{MLP}_{\text{lookup}} : \mathbb{R}^d \to \mathbb{R}^d$ such that, for $\boldsymbol{u}_t = \sum_k \alpha_{t,k}\boldsymbol{u}_{k,s_t}$,

$$\texttt{MLP}_{\text{lookup}}(\boldsymbol{u}_t) = \sum_{k=1}^{K} \alpha_{t,k}\boldsymbol{f}(k,s_t) = \boldsymbol{E}\sum_{k=1}^{K} \alpha_{t,k}\boldsymbol{p}_{k,s_t} - \boldsymbol{u}_t = \boldsymbol{E}\boldsymbol{q}_{t+1} - \boldsymbol{u}_t.$$

Composing the first-layer affine forms with $\boldsymbol{C}$, the map $\boldsymbol{x}_t \mapsto \texttt{MLP}_{\text{lookup}}(\boldsymbol{C}\boldsymbol{x}_t)$ is still a two-layer ReLU MLP with hidden width $KV$. Therefore the first residual block outputs

$$\boldsymbol{y}_t := \boldsymbol{x}_t + \texttt{MLP}_{\text{lookup}}(\boldsymbol{C}\boldsymbol{x}_t) = \boldsymbol{\mu}_t + \boldsymbol{u}_t + (\boldsymbol{E}\boldsymbol{q}_{t+1} - \boldsymbol{u}_t) = \boldsymbol{\mu}_t + \boldsymbol{E}\boldsymbol{q}_{t+1}.$$

Thus Block 1 does not cancel the nuisance component $\boldsymbol{\mu}_t$. It carries $\boldsymbol{\mu}_t$ forward and only converts the task/token component into the embedded Bayesian predictive $\boldsymbol{E}\boldsymbol{q}_{t+1}$.

**Step 2: Read out $q_{t+1}$, cancel only at the logits, and apply softmax.** Since $\mathrm{col}(\boldsymbol{E}) = \mathrm{span}\{\boldsymbol{\nu}_a : a \in \mathcal{V}\} \subset \mathcal{U}$ and $\mathcal{M}_{\mathcal{I}} \cap \mathcal{U} = \{0\}$, we have

$$\mathcal{M}_{\mathcal{I}} \cap \mathrm{col}(\boldsymbol{E}) = \{0\}.$$

By condition (iv), $\boldsymbol{q}_{t+1} \in \Delta_V^{\varepsilon}$. Applying Lemma 2 with $\mathcal{M} = \mathcal{M}_{\mathcal{I}}$ gives a constructed unembedding $\boldsymbol{W}_{\mathrm{U}} \in \mathbb{R}^{V\times d}$ and a width-$m_{\log}(\delta,\varepsilon,V) + 2V$ two-layer ReLU MLP $\texttt{MLP}_{\text{out}} : \mathbb{R}^d \to \mathbb{R}^d$ such that, for $\boldsymbol{y}_t = \boldsymbol{\mu}_t + \boldsymbol{E}\boldsymbol{q}_{t+1}$,

$$\left\|\mathrm{softmax}\big(\boldsymbol{W}_{\mathrm{U}}\big(\boldsymbol{y}_t + \texttt{MLP}_{\text{out}}(\boldsymbol{y}_t)\big)\big) - \boldsymbol{q}_{t+1}\right\|_{\infty} < \delta.$$

Using this MLP as the second block's MLP sublayer proves the claim. All maps used above, namely $\boldsymbol{C}$, $\boldsymbol{E}$, $\texttt{MLP}_{\text{lookup}}$, $\boldsymbol{W}_{\mathrm{U}}$, and $\texttt{MLP}_{\text{out}}$, are fixed across $t \in \mathcal{I}$, so the construction is uniform over positions. □

### B.2.2 Technical Lemmas and Proofs

This subsection states and proves the two technical lemmas invoked in the construction of Sec. B.2.1: Lemma 1 is used in the first appended block, and Lemma 2 is used in the second. Both isolate a single capability of a two-layer ReLU MLP and are stated in self-contained form, independent of the transformer setting.

**Lemma 1** (Width-$KV$ lookup and convex-linearity in the latent index)**.** *Let $K, V, d_{\text{out}} \in \mathbb{N}$. Suppose*

$$\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_V\} \subset \mathbb{R}^d$$

*is linearly independent, and define*

$$\boldsymbol{u}_{k,a} := \boldsymbol{\theta}_k + \boldsymbol{\nu}_a, \qquad (k,a) \in [K] \times [V].$$

*For any map $\boldsymbol{f} : [K] \times [V] \to \mathbb{R}^{d_{\text{out}}}$, there exists a two-layer ReLU MLP $L : \mathbb{R}^d \to \mathbb{R}^{d_{\text{out}}}$ with hidden width $KV$ such that*

$$L(\boldsymbol{u}_{k,a}) = \boldsymbol{f}(k,a)$$

*for all $(k,a)$, and, more generally,*

$$L\left(\sum_{k=1}^{K} \alpha_k \boldsymbol{u}_{k,a}\right) = \sum_{k=1}^{K} \alpha_k \boldsymbol{f}(k,a)$$

*for every fixed $a \in [V]$ and every $\boldsymbol{\alpha} \in \Delta^{K-1} := \{\boldsymbol{\alpha} \in \mathbb{R}^K : \alpha_k \geq 0, \sum_k \alpha_k = 1\}$.*

*Proof.* By linear independence, there is a biorthogonal dual family

$$\{\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_K^*, \boldsymbol{\nu}_1^*, \ldots, \boldsymbol{\nu}_V^*\} \subset \mathbb{R}^d$$

satisfying

$$\langle \boldsymbol{\theta}_i^*, \boldsymbol{\theta}_k \rangle = \mathbf{1}\{i = k\}, \qquad \langle \boldsymbol{\theta}_i^*, \boldsymbol{\nu}_a \rangle = 0,$$

and

$$\langle \boldsymbol{\nu}_j^*, \boldsymbol{\nu}_a \rangle = \mathbf{1}\{j = a\}, \qquad \langle \boldsymbol{\nu}_j^*, \boldsymbol{\theta}_k \rangle = 0.$$

For each pair $(i, j) \in [K] \times [V]$, define

$$\boldsymbol{\phi}_{ij} := \boldsymbol{\theta}_i^* + \boldsymbol{\nu}_j^*.$$

Then

$$\langle \boldsymbol{\phi}_{ij}, \boldsymbol{u}_{k,a} \rangle = \mathbf{1}\{i = k\} + \mathbf{1}\{j = a\}.$$

Define one hidden unit per pair:

$$h_{ij}(\boldsymbol{u}) := \sigma\big(\langle \boldsymbol{\phi}_{ij}, \boldsymbol{u} \rangle - 1\big), \qquad \sigma(r) := \max\{r, 0\}.$$

On the vertices $\boldsymbol{u}_{k,a}$, this gives

$$h_{ij}(\boldsymbol{u}_{k,a}) = \mathbf{1}\{i = k,\, j = a\}.$$

Now fix a token $a$ and a simplex vector $\boldsymbol{\alpha}$. For

$$\boldsymbol{u}_a^{(\boldsymbol{\alpha})} := \sum_{k=1}^{K} \alpha_k \boldsymbol{u}_{k,a} = \sum_{k=1}^{K} \alpha_k \boldsymbol{\theta}_k + \boldsymbol{\nu}_a,$$

we have

$$\langle \boldsymbol{\phi}_{ij}, \boldsymbol{u}_a^{(\boldsymbol{\alpha})} \rangle = \alpha_i + \mathbf{1}\{j = a\}.$$

Therefore,

$$h_{ij}(\boldsymbol{u}_a^{(\boldsymbol{\alpha})}) = \begin{cases} \sigma(\alpha_i) = \alpha_i, & j = a, \\ \sigma(\alpha_i - 1) = 0, & j \neq a, \end{cases}$$

because $0 \leq \alpha_i \leq 1$. Finally define

$$L(\boldsymbol{u}) := \sum_{i=1}^{K} \sum_{j=1}^{V} h_{ij}(\boldsymbol{u})\, \boldsymbol{f}(i, j).$$

This is a two-layer ReLU MLP with hidden width $KV$, output weights $\boldsymbol{f}(i, j)$, and zero output bias. The vertex and convex-linearity claims follow from the displayed identities. □

**Remark 3** (Connection to key–value memory)**.** *The construction above can be interpreted through the lens of key–value memory, as discussed in Geva et al. [18]. In our setting, the vectors $\{\boldsymbol{\phi}_{ij}\}$ play the role of keys, which detect whether the input corresponds to a specific pair $(i, j)$ via inner products and thresholding. The second layer associates each key with a value $\boldsymbol{f}(i, j)$, and the MLP output is obtained by aggregating the retrieved values.*

**Lemma 2** (Probability recovery with a constructed unembedding)**.** *Let $\mathcal{M} \subset \mathbb{R}^d$ be a linear subspace, let $\boldsymbol{E} \in \mathbb{R}^{d \times V}$ have full column rank, and assume*

$$\mathcal{M} \cap \mathrm{col}(\boldsymbol{E}) = \{0\}.$$

*For every $\varepsilon \in (0, 1/V)$ and $\delta > 0$, there exist an output unembedding $\boldsymbol{W}_{\mathrm{U}} \in \mathbb{R}^{V \times d}$ and a two-layer ReLU MLP $F : \mathbb{R}^d \to \mathbb{R}^d$ of hidden width $m_{\log}(\delta, \varepsilon, V) + 2V$ such that*

$$\sup_{\boldsymbol{m} \in \mathcal{M},\, \boldsymbol{p} \in \Delta_V^{\varepsilon}} \big\| \mathrm{softmax}\big(\boldsymbol{W}_{\mathrm{U}}\big(\boldsymbol{m} + \boldsymbol{E}\boldsymbol{p} + F(\boldsymbol{m} + \boldsymbol{E}\boldsymbol{p})\big)\big) - \boldsymbol{p} \big\|_{\infty} < \delta,$$

*where*

$$\Delta_V^{\varepsilon} := \left\{ \boldsymbol{p} \in \mathbb{R}^V : p_b \geq \varepsilon \text{ for all } b \in [V],\ \sum_{b=1}^{V} p_b = 1 \right\}.$$

*Proof.* Because $M \cap \mathrm{col}(\boldsymbol{E}) = \{0\}$, we have $M + \mathrm{col}(\boldsymbol{E}) = M \oplus \mathrm{col}(\boldsymbol{E})$. Thus every vector in this sum has a unique decomposition $\boldsymbol{m} + \boldsymbol{E}\boldsymbol{r}$ with $\boldsymbol{m} \in M$ and $\boldsymbol{r} \in \mathbb{R}^V$. Define a linear map $\boldsymbol{P} : M \oplus \mathrm{col}(\boldsymbol{E}) \to \mathbb{R}^V$ by

$$\boldsymbol{P}(\boldsymbol{m} + \boldsymbol{E}\boldsymbol{r}) = \boldsymbol{r}.$$

Extend $\boldsymbol{P}$ linearly to all of $\mathbb{R}^d$, and set

$$\boldsymbol{W}_{\mathrm{U}} := \boldsymbol{P}.$$

Then

$$\boldsymbol{W}_{\mathrm{U}}\boldsymbol{m} = \boldsymbol{0} \quad (\boldsymbol{m} \in \mathcal{M}), \qquad \boldsymbol{W}_{\mathrm{U}}\boldsymbol{E} = \boldsymbol{I}_V.$$

In particular, the required unembedding is constructed from the decomposition; no pre-existing output head is assumed.

Choose $\eta > 0$ such that $e^{2\eta} - 1 < \delta$. Let $G : \mathbb{R}^V \to \mathbb{R}^V$ be a two-layer ReLU network satisfying

$$\sup_{\boldsymbol{p} \in \Delta_V^{\varepsilon}} \|G(\boldsymbol{p}) - \log \boldsymbol{p}\|_\infty < \eta.$$

Such a $G$ can be obtained coordinatewise by a piecewise-linear ReLU spline approximation to $\log$ on the compact interval $[\varepsilon, 1]$; equivalently, this follows from standard single-hidden-layer approximation results such as Pinkus [37]. Let $m_{\log}(\delta, \varepsilon, V)$ denote the width of this coordinatewise approximation.

Define

$$F(\boldsymbol{y}) := \boldsymbol{E}\, G(\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y}) - \boldsymbol{E}\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y}.$$

The first term is a two-layer ReLU MLP of width $m_{\log}(\delta, \varepsilon, V)$, obtained by composing the input layer of $G$ with $\boldsymbol{W}_{\mathrm{U}}$ and the output layer with $\boldsymbol{E}$. The second term $-\boldsymbol{E}\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y}$ is a rank-at-most-$V$ linear map, so it can be represented exactly using at most $2V$ ReLU hidden units. Indeed, if $\boldsymbol{E}\boldsymbol{W}_{\mathrm{U}} = \boldsymbol{A}\boldsymbol{B}$ with $\boldsymbol{B} \in \mathbb{R}^{r \times d}$, $\boldsymbol{A} \in \mathbb{R}^{d \times r}$, and $r \le V$, then

$$-\boldsymbol{E}\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y} = -\boldsymbol{A}\big(\sigma(\boldsymbol{B}\boldsymbol{y}) - \sigma(-\boldsymbol{B}\boldsymbol{y})\big).$$

Concatenating the hidden units for the two terms gives a two-layer ReLU MLP of width $m_{\log}(\delta, \varepsilon, V) + 2V$.

For $\boldsymbol{y} = \boldsymbol{m} + \boldsymbol{E}\boldsymbol{p}$ with $\boldsymbol{m} \in \mathcal{M}$ and $\boldsymbol{p} \in \Delta_V^{\varepsilon}$, we have $\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y} = \boldsymbol{p}$. Therefore

$$\boldsymbol{W}_{\mathrm{U}}\big(\boldsymbol{y} + F(\boldsymbol{y})\big) = \boldsymbol{W}_{\mathrm{U}}\boldsymbol{y} + \boldsymbol{W}_{\mathrm{U}}\boldsymbol{E}\, G(\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y}) - \boldsymbol{W}_{\mathrm{U}}\boldsymbol{E}\boldsymbol{W}_{\mathrm{U}}\boldsymbol{y} = G(\boldsymbol{p}),$$

where we used $\boldsymbol{W}_{\mathrm{U}}\boldsymbol{E} = \boldsymbol{I}_V$. The output distribution is therefore $\mathrm{softmax}(G(\boldsymbol{p}))$. Write $G(\boldsymbol{p}) = \log \boldsymbol{p} + \boldsymbol{e}$ with $\|\boldsymbol{e}\|_\infty < \eta$. For each coordinate $b$,

$$\mathrm{softmax}(G(\boldsymbol{p}))_b = \frac{p_b e^{e_b}}{\sum_{c=1}^{V} p_c e^{e_c}}.$$

Since $e^{-\eta} \le e^{e_c} \le e^{\eta}$ for every $c$, the ratio $\mathrm{softmax}(G(\boldsymbol{p}))_b / p_b$ lies in $[e^{-2\eta}, e^{2\eta}]$. Thus

$$|\mathrm{softmax}(G(\boldsymbol{p}))_b - p_b| \le e^{2\eta} - 1 < \delta.$$

Taking the maximum over $b$ proves the lemma. □

### B.3 Details About Extrapolative Task Learning Mode: Explicit Formulas

This subsection derives the closed-form predictors that define the extrapolative task-learning mode `M2` introduced in Sec. 4.3. Concretely, `M2` is the Bayesian posterior predictive obtained from Eq. 6 by replacing the prior $\pi_{\mathrm{train}}$ over the finite training support $\mathcal{Z}_{\mathrm{train}}$ with the *generative* prior over the full latent space $\mathcal{Z}$, i.e. the same parametric distribution that was used to sample tasks during training (symmetric Dirichlet for `E1` and `E3`, isotropic Gaussian for `E2`; cf. Sec. 3). For each of `E1`–`E3`, conjugacy of this prior with the per-token likelihood collapses the integral over $\mathcal{Z}$ into a closed-form predictor that depends on the observed sequence only through a low-dimensional context statistic: a smoothed empirical unigram for `E1`, the ridge solution for `E2`, and a smoothed empirical bigram for `E3`. The derivations below thus pin down the concrete predictive targets that a model implementing `M2` must encode; whether the trained model actually encodes them, and where in the residual stream, is a separate empirical question taken up in the orthogonal-subspace decomposition of Sec. D.10.

**Biased dice (`E1`).** Each latent task $z = \boldsymbol{p} \in \mathcal{Z} = \Delta^{V-1}$ is sampled from the symmetric Dirichlet prior $\mathrm{Dir}(\boldsymbol{1}_V)$. More generally, for $\boldsymbol{p} \in \Delta^{V-1}$ drawn from $\mathrm{Dir}(\alpha_0 \boldsymbol{1}_V)$, the probability density function is

$$f\,(p_1, \ldots, p_V; \alpha_0) = \frac{\Gamma(\alpha_0 V)}{\Gamma(\alpha_0)^V} \prod_{a=1}^{V} p_a^{\alpha_0 - 1}.$$

Recall the token count $n_a(t)$ defined in Equation (18). The posterior predictive after observing $s_{\le t}$ is

$$\begin{aligned}
\mathbb{P}(s_{t+1}=a \mid s_{\le t}) &= \int_{\boldsymbol{p}\in\mathcal{Z}} \mathbb{P}(s_{t+1}=a \mid \boldsymbol{p}, s_{\le t})\,\mathbb{P}(\boldsymbol{p} \mid s_{\le t})\,d\boldsymbol{p} \\
&= \frac{\int_{\boldsymbol{p}\in\mathcal{Z}} \mathbb{P}(s_{t+1}=a \mid \boldsymbol{p})\,\mathbb{P}(s_{\le t} \mid \boldsymbol{p})\,\mathbb{P}(\boldsymbol{p})\,d\boldsymbol{p}}{\int_{\boldsymbol{p}\in\mathcal{Z}} \mathbb{P}(s_{\le t} \mid \boldsymbol{p})\,\mathbb{P}(\boldsymbol{p})\,d\boldsymbol{p}} \\
&= \frac{\int_{\boldsymbol{p}\in\mathcal{Z}} p_a \prod_{b=1}^{V} p_b^{n_b(t)+\alpha_0-1}\,d\boldsymbol{p}}{\int_{\boldsymbol{p}\in\mathcal{Z}} \prod_{b=1}^{V} p_b^{n_b(t)+\alpha_0-1}\,d\boldsymbol{p}} \\
&= \frac{\Gamma(n_a(t)+\alpha_0+1)\prod_{b\neq a}^{V}\Gamma(n_b(t)+\alpha_0)}{\Gamma(t+V\alpha_0+1)} \cdot \frac{\Gamma(t+V\alpha_0)}{\prod_{b=1}^{V}\Gamma(n_b(t)+\alpha_0)} \\
&= \frac{n_a(t)+\alpha_0}{t+V\alpha_0},
\end{aligned}$$

which is a smoothed empirical unigram estimator.

**Noisy linear regression (E2).** Each latent task $z=\boldsymbol{w}\in\mathcal{Z}=\mathbb{R}^D$ is drawn from an isotropic Gaussian prior

$$\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\,\tau^{-2}\sigma^2\boldsymbol{I}_D),$$

which matches the generative prior of Sec. 3 when $\tau^2=\sigma^2$. Conditional on $\boldsymbol{w}$, inputs are i.i.d. $\boldsymbol{x}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_D)$ and $y_t\mid\boldsymbol{x}_t,\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{x}_t^\top\boldsymbol{w},\,\sigma^2)$. For the prediction of $y_{t+1}$ at a fresh covariate $\boldsymbol{x}_{t+1}$, let $\boldsymbol{X}_{\le t}=(\boldsymbol{x}_1,\dots,\boldsymbol{x}_t)^\top\in\mathbb{R}^{t\times D}$ and $\boldsymbol{y}_{\le t}=(y_1,\dots,y_t)^\top\in\mathbb{R}^t$ collect the observed pairs. By Gaussian–Gaussian conjugacy, the posterior over $\boldsymbol{w}$ is again Gaussian:

$$\boldsymbol{w}\mid\boldsymbol{X}_{\le t},\boldsymbol{y}_{\le t}\;\sim\;\mathcal{N}\big(\hat{\boldsymbol{w}}_{\mathrm{ridge}}(t),\;\sigma^2\,\boldsymbol{\Sigma}_t\big)\,,\qquad \boldsymbol{\Sigma}_t:=\big(\boldsymbol{X}_{\le t}^\top\boldsymbol{X}_{\le t}+\tau^2\boldsymbol{I}_D\big)^{-1},$$

where $\hat{\boldsymbol{w}}_{\mathrm{ridge}}(t)$ is the ridge solution [59, 40, 2]

$$\hat{\boldsymbol{w}}_{\mathrm{ridge}}(t)\;=\;\big(\boldsymbol{X}_{\le t}^\top\boldsymbol{X}_{\le t}+\tau^2\boldsymbol{I}_D\big)^{-1}\boldsymbol{X}_{\le t}^\top\boldsymbol{y}_{\le t}.$$

Marginalising $\boldsymbol{w}$ yields the posterior predictive

$$\begin{aligned}
\mathbb{P}\big(y_{t+1}\mid\boldsymbol{x}_{t+1},\boldsymbol{X}_{\le t},\boldsymbol{y}_{\le t}\big) &= \int_{\mathbb{R}^D}\mathbb{P}(y_{t+1}\mid\boldsymbol{x}_{t+1},\boldsymbol{w})\,\mathbb{P}(\boldsymbol{w}\mid\boldsymbol{X}_{\le t},\boldsymbol{y}_{\le t})\,d\boldsymbol{w} \\
&= \mathcal{N}\big(\boldsymbol{x}_{t+1}^\top\hat{\boldsymbol{w}}_{\mathrm{ridge}}(t),\;\sigma^2\big(1+\boldsymbol{x}_{t+1}^\top\boldsymbol{\Sigma}_t\,\boldsymbol{x}_{t+1}\big)\big)\,.
\end{aligned}$$

Hence the posterior predictive mean is the ridge regressor evaluated at $\boldsymbol{x}_{t+1}$, and the predictive variance inflates the noise level $\sigma^2$ by the input-dependent factor $1+\boldsymbol{x}_{t+1}^\top\boldsymbol{\Sigma}_t\boldsymbol{x}_{t+1}$ that captures remaining uncertainty about $\boldsymbol{w}$.

**Mixture of Markov chains (E3).** Each latent task $z=\boldsymbol{P}\in\mathcal{Z}=(\Delta^{V-1})^V$ is a row-stochastic $V\times V$ transition matrix, with each row drawn independently from the symmetric Dirichlet prior used to generate the training tasks (Sec. 3):

$$\boldsymbol{P}_{a,\cdot}\overset{\text{i.i.d.}}{\sim}\mathrm{Dir}(\alpha_0\boldsymbol{1}_V),\qquad a=1,\dots,V.$$

For a Markov sequence $s_{\le t}$ generated under $\boldsymbol{P}$ from a uniform initial distribution, let

$$N_{ab}(t)\;:=\;\sum_{\tau=2}^{t}\mathbf{1}\{s_{\tau-1}=a,\,s_\tau=b\}\;\in\;\mathbb{N},\qquad N_{a\cdot}(t)\;:=\;\sum_{b=1}^{V}N_{ab}(t)\;\in\;\mathbb{N},$$

denote the count of $a\to b$ transitions and the row total, respectively. The likelihood factorises across rows,

$$\mathbb{P}(s_{\le t}\mid\boldsymbol{P})\;=\;\prod_{a=1}^{V}\prod_{b=1}^{V}P_{a,b}^{N_{ab}(t)},$$

which is a product of multinomial likelihoods. Combined with the row-independent Dirichlet prior, Dirichlet–multinomial conjugacy gives a posterior that factorises across rows,

$$\boldsymbol{P}_{a,\cdot}\mid s_{\le t}\;\sim\;\mathrm{Dir}(\alpha_0+N_{a1}(t),\,\dots,\,\alpha_0+N_{aV}(t))\,,\qquad a=1,\dots,V.$$

The posterior predictive at the next position, given current state $s_t = a$, is then

$$\begin{aligned}\mathbb{P}\big(s_{t+1} = b \mid s_t = a,\, s_{\le t}\big) &= \int_{\mathcal{Z}} P_{a,b}\, \mathbb{P}(\boldsymbol{P} \mid s_{\le t})\, d\boldsymbol{P} \\ &= \mathbb{E}[P_{a,b} \mid s_{\le t}] \\ &= \frac{N_{ab}(t) + \alpha_0}{N_{a\cdot}(t) + V\alpha_0},\end{aligned}$$

where the last equality uses the mean of a Dirichlet distribution. This is a smoothed empirical bigram estimator, exactly analogous to the smoothed empirical unigram derived for `E1`: it pools every observed transition out of state $a$ and interpolates with the uniform prior at rate controlled by $\alpha_0$.

**Summary.** In all three experiments, the predictor underlying `M2` is the exact Bayesian posterior predictive under the parametric prior used to generate the training tasks, but evaluated over the *full* latent space $\mathcal{Z}$ rather than only the finite training support $\mathcal{Z}_{\text{train}}$. Conjugacy of each prior–likelihood pair (Dirichlet–categorical for `E1`, Gaussian–Gaussian for `E2`, and row-wise Dirichlet–multinomial for `E3`) yields a closed-form predictor that depends on the observed sequence only through a sufficient context statistic: the empirical unigram counts $n_a(t)$ for `E1`, the ridge solution $\hat{\boldsymbol{w}}_{\text{ridge}}(t)$ for `E2`, and the empirical bigram counts $N_{ab}(t)$ for `E3`. This explains why these context statistics arise naturally as the targets that the model must encode in order to implement `M2`.

### B.4 Proof of Theorem 2 in Section 4.3

This subsection states and proves the formal version of Theorem 2. The result makes precise the intuition that the extrapolative task-learning mode `M2` cannot be implemented from inside a task-vector subspace whose dimension $k_\star$ is smaller than the intrinsic dimension $d_0$ of the optimal prediction map: a volumetric packing argument forbids the required separation of latents within a $k_\star$-dimensional affine slab unless the alignment error $\varepsilon$ grows accordingly. Theorem 4 below gives the explicit lower bound on $\varepsilon$, which informally takes the form $\varepsilon \gtrsim (c-\delta)/L - C/(2^{d_0/k_\star} - 1)$ reported in Sec. 4.3.

**Remark 4** (Scope of Theorem 2)**.** *The theorem is a packing obstruction: the extrapolative predictive family has intrinsic dimension $d_0 \gg k_\star$, so a Lipschitz decoder near the task simplex cannot realize that many predictions, and `M2` representations are forced away from the task subspace. However, it only establishes a separation result, without forcing the geometry of the exit to be orthogonal. Sec. 6 shows empirically that the exit is near-orthogonal and that this geometry is learned during training rather than inherited from initialization. Sec. D.11 offers a hypothetical explanation: under limited shared dimensions, near-orthogonal placement minimizes interference between retrieval and extrapolation [57, 16].*

**Notation.** We collect the notation used throughout this appendix, extending the conventions of Sec. 4. Let $\mathcal{Z}$ denote the latent space with training set $\mathcal{Z}_{\text{train}} = [K]$. At a fixed layer, $\boldsymbol{h}_t \in \mathbb{R}^d$ is the hidden state at position $t$; $\{\boldsymbol{\theta}_k\}_{k\le K}$ and $\{\boldsymbol{\nu}_a\}_{a\in\mathcal{V}}$ are the task vectors and token-encoding vectors (Definition 1); $\boldsymbol{\mu}_{t,k,a} := \mathbb{E}[\boldsymbol{h}_t \mid z{=}k,\, s_t{=}a]$ are the cell means (Eq. 1). We assume `P0` together with the existence of the limiting cell mean $\boldsymbol{\mu}(z,a) := \lim_{t\to\infty} \boldsymbol{\mu}_{t,z,a}$ for each $(z,a)$ in the neighborhood of interest. (Note that `P0` alone controls only the conditional residual variance $\mathbb{E}\|\boldsymbol{h}_t - \boldsymbol{\mu}_{t,z,s_t}\|^2$ and does not by itself guarantee convergence of $\boldsymbol{\mu}_{t,z,a}$; we therefore impose the existence of the limit explicitly.) Under these assumptions the (long-context) decoder maps $\boldsymbol{\mu}(z,a)$ to the Bayes-optimal next-token distribution $\mathbb{P}(\cdot \mid z,a)$.

**Setup.** We now introduce the symbols specific to the simplex-confinement question, augmenting the notation of Sec. 4.3. Fix an anchor token $a_\star \in \mathcal{V}$ and write

$$\boldsymbol{p}(z) := \mathbb{P}(\cdot \mid z, a_\star) \in \Delta^{V-1}, \qquad \boldsymbol{\mu}_\star(z) := \boldsymbol{\mu}(z, a_\star) \in \mathbb{R}^d,$$

for the conditional next-token distribution and the limiting cell mean at $a_\star$. Let

$$\mathcal{T} := \operatorname{span}(\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K - \boldsymbol{\theta}_1), \qquad \mathcal{S} := \boldsymbol{\nu}_{a_\star} + \mathcal{T},$$

denote the task-vector subspace and its affine translate at $\boldsymbol{\nu}_{a_\star}$, and set $k_\star := \dim(\mathcal{T}) \le K-1$. We assume $k_\star \ge 1$, as otherwise $\mathcal{S}$ is a singleton and the conclusions are trivial.

**Discrete-output scope.** The theorem is stated for finite discrete output spaces: $a_\star \in \mathcal{V}$ and $\boldsymbol{p}(z) = \mathbb{P}(\cdot \mid z, a_\star) \in \Delta^{V-1}$. It therefore directly covers the discrete-output settings `E1` and `E3`. The linear-regression setting E2 is an empirical continuous-output analogue. A formal continuous-output version would require replacing the simplex-valued map $\boldsymbol{p}$ by an appropriate notion of distance between posterior-predictive regression distributions. We omit this extension for simplicity.

**Gauge convention** The affine anchor in $\mathcal{S} = \boldsymbol{\nu}_{a_\star} + \mathcal{T}$ is a gauge choice. The proof below uses only the affine set $\mathcal{S}$ and distances $\operatorname{dist}(\boldsymbol{\mu}_\star(z), \mathcal{S})$, not the separate decomposition of its anchor into global-mean and token components. Indeed, if `P1` is written in the centered gauge

$$\boldsymbol{h}_t \approx \boldsymbol{\mu}_t + \boldsymbol{\theta}_z + \boldsymbol{\nu}_{s_t}$$

and $\boldsymbol{\mu}_t \to \boldsymbol{\mu}_\infty$, then the long-context ID cell means at anchor token $a_\star$ lie in

$$\boldsymbol{\mu}_\infty + \boldsymbol{\nu}_{a_\star} + \mathcal{T}.$$

We absorb this common translate into the affine anchor, equivalently writing

$$\bar{\boldsymbol{\nu}}_{a_\star} := \boldsymbol{\mu}_\infty + \boldsymbol{\nu}_{a_\star}, \qquad \mathcal{S} = \bar{\boldsymbol{\nu}}_{a_\star} + T,$$

and then drop the bar. This is only a notational convention for interpreting $\mathcal{S}$ under `P1`; Theorem 2 itself assumes only the limiting cell means $\boldsymbol{\mu}(z, a)$ specified above.

**Definition 2** (Local regularity setup)**.** *Fix a base point $z_0 \in \operatorname{int}(\mathcal{Z})$ and assume that $\boldsymbol{p}$ is $C^1$ on a neighborhood of $z_0$. Let*

$$\boldsymbol{J}_0 := \nabla_z \boldsymbol{p}(z_0), \qquad d_0 := \operatorname{rank}(\boldsymbol{J}_0) \geq 1, \qquad \sigma_0 := s_{d_0}(\boldsymbol{J}_0) > 0, \qquad \mathcal{E} := (\ker \boldsymbol{J}_0)^\perp,$$

*where $s_{d_0}(\boldsymbol{J}_0)$ denotes the smallest nonzero singular value of $\boldsymbol{J}_0$. Choose a radius $r_0 > 0$ with $B(z_0, r_0) \subseteq \mathcal{Z}$, and define*

$$U_{\mathcal{E},0} := (z_0 + \mathcal{E}) \cap B(z_0, r_0).$$

*Then $U_{\mathcal{E},0}$ is a $d_0$-dimensional ball in the affine subspace $z_0 + \mathcal{E}$ along which $\boldsymbol{p}$ has full local rank; the integer $d_0$ makes precise the "number of independent directions along which $\boldsymbol{p}(z)$ varies" from Sec. 4.3.*

The next theorem formalizes the informal claim of Theorem 2 in Sec. 4.3. Throughout, we write

$$\operatorname{dist}(\boldsymbol{x}, \mathcal{S}) := \inf_{\boldsymbol{c} \in \mathcal{S}} \|\boldsymbol{x} - \boldsymbol{c}\|_2$$

for the Euclidean distance from a point $\boldsymbol{x} \in \mathbb{R}^d$ to the affine set $\mathcal{S}$, and we collect the regularity assumptions used in the theorem.

**Definition 3** (Local decoder regularity)**.** *Let $\hat{\boldsymbol{p}}(z)$ be the predictor obtained by passing $\boldsymbol{\mu}_\star(z)$ through the model's decoder. We say the setup is* $(M, L, \delta)$-regular on a neighborhood $\mathcal{U} \subseteq \mathcal{Z}$ *of $z_0$ if there exist constants $M < \infty$, $L > 0$, and $\delta \geq 0$ such that, for all $z, z' \in \mathcal{U}$,*

$$\|\boldsymbol{\mu}_\star(z) - \boldsymbol{\mu}_\star(z')\|_2 \leq M, \tag{7}$$

$$\|\hat{\boldsymbol{p}}(z) - \hat{\boldsymbol{p}}(z')\|_2 \leq L \, \|\boldsymbol{\mu}_\star(z) - \boldsymbol{\mu}_\star(z')\|_2, \tag{8}$$

$$\|\hat{\boldsymbol{p}}(z) - \boldsymbol{p}(z)\|_2 \leq \delta. \tag{9}$$

*We further define the worst-case distance from $\boldsymbol{\mu}_\star$ to the affine task subspace on $\mathcal{U}$,*

$$\varepsilon := \sup_{z \in \mathcal{U}} \operatorname{dist}\big(\boldsymbol{\mu}_\star(z), \mathcal{S}\big). \tag{10}$$

**Theorem 4** (Finite-context packing obstruction)**.** *Assume the setup of Definition 2 and that the $(M, L, \delta)$-regularity of Definition 3 holds on a neighborhood $\mathcal{U} \subseteq \mathcal{Z}$ of $z_0$. Then there exists $r_{\mathrm{loc}} \in (0, r_0]$ such that*

$$U_{\mathcal{E}} := (z_0 + \mathcal{E}) \cap B(z_0, r_{\mathrm{loc}}) \subseteq \mathcal{U}, \qquad \alpha := \sigma_0/2,$$

*and the following hold.*

***(a) Local co-Lipschitz estimate for the ground-truth map.*** *For all $z, z' \in U_{\mathcal{E}}$,*

$$\|\boldsymbol{p}(z) - \boldsymbol{p}(z')\|_2 \geq \alpha \, \|z - z'\|_2. \tag{11}$$

***(b) Finite-context packing inequality.*** *For any $0 < \rho < r_{\mathrm{loc}}$ such that*

$$\frac{\alpha\rho - 2\delta}{L} > 2\varepsilon,$$

*one has*

$$\left(\frac{r_{\mathrm{loc}}}{\rho}\right)^{d_0} \le \left(1 + \frac{2M}{\sqrt{\left(\frac{\alpha\rho - 2\delta}{L}\right)^2 - 4\varepsilon^2}}\right)^{k_\star}. \tag{12}$$

***(c) Lower bound on the confinement radius.*** *For any $0 < \rho < r_{\mathrm{loc}}$ such that $\alpha\rho > 2\delta$,*

$$\varepsilon \ge \frac{1}{2}\left[\left(\frac{\alpha\rho - 2\delta}{L}\right)^2 - \left(\frac{2M}{(r_{\mathrm{loc}}/\rho)^{d_0/k_\star} - 1}\right)^2\right]_+^{1/2}, \tag{13}$$

*where $[x]_+ := \max\{x, 0\}$.*

*Proof.* Choose a linear isometry

$$\boldsymbol{W} : \mathbb{R}^{d_0} \to \mathcal{E}.$$

Define

$$\widetilde{\boldsymbol{p}}(\boldsymbol{u}) := \boldsymbol{p}(z_0 + \boldsymbol{W}\boldsymbol{u}), \qquad \boldsymbol{u} \in B_{\mathbb{R}^{d_0}}(\mathbf{0}, r_0).$$

Then $\widetilde{\boldsymbol{p}}$ is $C^1$, and

$$\nabla\widetilde{\boldsymbol{p}}(\mathbf{0}) = \boldsymbol{J}_0\boldsymbol{W}.$$

Since $\boldsymbol{W}$ is an isometry onto $\mathcal{E} = (\ker \boldsymbol{J}_0)^\perp$, the smallest singular value of $\nabla\widetilde{\boldsymbol{p}}(\mathbf{0})$ equals $\sigma_0$. By continuity of $\nabla\widetilde{\boldsymbol{p}}$ at $\mathbf{0}$, after shrinking $r_{\mathrm{loc}} \in (0, r_0]$ if necessary, we may assume both that $U_{\mathcal{E}} \subseteq \mathcal{U}$ and that

$$\sup_{\|\boldsymbol{u}\|_2 < r_{\mathrm{loc}}} \|\nabla\widetilde{\boldsymbol{p}}(\boldsymbol{u}) - \nabla\widetilde{\boldsymbol{p}}(\mathbf{0})\|_{\mathrm{op}} \le \frac{\sigma_0}{2}.$$

Now fix $\boldsymbol{u}, \boldsymbol{v} \in B_{\mathbb{R}^{d_0}}(\mathbf{0}, r_{\mathrm{loc}})$. By the fundamental theorem of calculus,

$$\widetilde{\boldsymbol{p}}(\boldsymbol{u}) - \widetilde{\boldsymbol{p}}(\boldsymbol{v}) = \nabla\widetilde{\boldsymbol{p}}(\mathbf{0})(\boldsymbol{u} - \boldsymbol{v}) + \int_0^1 \Big(\nabla\widetilde{\boldsymbol{p}}\big(\boldsymbol{v} + \tau(\boldsymbol{u} - \boldsymbol{v})\big) - \nabla\widetilde{\boldsymbol{p}}(\mathbf{0})\Big)(\boldsymbol{u} - \boldsymbol{v})\, d\tau.$$

Therefore

$$\begin{aligned}\|\widetilde{\boldsymbol{p}}(\boldsymbol{u}) - \widetilde{\boldsymbol{p}}(\boldsymbol{v})\|_2 &\ge \|\nabla\widetilde{\boldsymbol{p}}(\mathbf{0})(\boldsymbol{u} - \boldsymbol{v})\|_2 - \frac{\sigma_0}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2 \\ &\ge \frac{\sigma_0}{2}\|\boldsymbol{u} - \boldsymbol{v}\|_2.\end{aligned}$$

Transporting this estimate back through $\boldsymbol{W}$ gives (11). This proves part (a).

Fix $0 < \rho < r_{\mathrm{loc}}$, and let

$$\{z_1, \dots, z_N\} \subset U_{\mathcal{E}}$$

be a maximal $\rho$-separated subset. Since the set is maximal,

$$U_{\mathcal{E}} \subseteq \bigcup_{i=1}^{N} \big((z_0 + \mathcal{E}) \cap B(z_i, \rho)\big).$$

Taking $d_0$-dimensional Euclidean volumes inside the affine space $z_0 + \mathcal{E}$, the standard ball-volume comparison (e.g. Corollary 4.2.13 in [53]) yields

$$N \ge \left(\frac{r_{\mathrm{loc}}}{\rho}\right)^{d_0}. \tag{14}$$

For $i \neq j$, (11) and (9) give

$$\begin{aligned}\|\hat{\boldsymbol{p}}(z_i) - \hat{\boldsymbol{p}}(z_j)\|_2 &\geq \|\boldsymbol{p}(z_i) - \boldsymbol{p}(z_j)\|_2 - \|\hat{\boldsymbol{p}}(z_i) - \boldsymbol{p}(z_i)\|_2 - \|\hat{\boldsymbol{p}}(z_j) - \boldsymbol{p}(z_j)\|_2 \\ &\geq \alpha\|z_i - z_j\|_2 - 2\delta \\ &\geq \alpha\rho - 2\delta.\end{aligned}$$

Using (8), we obtain

$$\|\boldsymbol{\mu}_\star(z_i) - \boldsymbol{\mu}_\star(z_j)\|_2 \geq \lambda(\rho), \qquad \lambda(\rho) := \frac{\alpha\rho - 2\delta}{L}.$$

Let $\Pi_{\mathcal{S}}$ be the orthogonal projection onto $\mathcal{S}$, and set

$$\boldsymbol{y}_i := \Pi_{\mathcal{S}}\boldsymbol{\mu}_\star(z_i) \in \mathcal{S}, \qquad \boldsymbol{u}_i := \boldsymbol{\mu}_\star(z_i) - \boldsymbol{y}_i \in \mathcal{T}^\perp.$$

By (10),

$$\|\boldsymbol{u}_i\|_2 = \operatorname{dist}\big(\boldsymbol{\mu}_\star(z_i), \mathcal{S}\big) \leq \varepsilon.$$

Since $\boldsymbol{y}_i - \boldsymbol{y}_j \in \mathcal{T}$ and $\boldsymbol{u}_i - \boldsymbol{u}_j \in \mathcal{T}^\perp$, the Pythagorean theorem gives

$$\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 = \|\boldsymbol{\mu}_\star(z_i) - \boldsymbol{\mu}_\star(z_j)\|_2^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_j\|_2^2 \geq \lambda(\rho)^2 - 4\varepsilon^2.$$

If $\lambda(\rho) > 2\varepsilon$, the points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are therefore $\Delta(\rho) := \sqrt{\lambda(\rho)^2 - 4\varepsilon^2}$-separated in $\mathcal{S}$.

Setting $\boldsymbol{c}_0 := \Pi_{\mathcal{S}}\boldsymbol{\mu}_\star(z_0)$, the 1-Lipschitz property of $\Pi_{\mathcal{S}}$ and (7) give $\|\boldsymbol{y}_i - \boldsymbol{c}_0\|_2 \leq M$, so

$$\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\} \subseteq \mathcal{S} \cap B(\boldsymbol{c}_0, M).$$

Assume now that $\lambda(\rho) > 2\varepsilon$. The balls $\mathcal{S} \cap B(\boldsymbol{y}_i, \Delta(\rho)/2)$, $i = 1, \ldots, N$, are pairwise disjoint and contained in $\mathcal{S} \cap B(\boldsymbol{c}_0, M + \Delta(\rho)/2)$. Comparing $k_\star$-dimensional Euclidean ball volumes once more (Corollary 4.2.13 in [53]) gives

$$N \leq \big(1 + 2M/\Delta(\rho)\big)^{k_\star}.$$

Combining with (14) yields (12), proving part (b).

Finally, fix $0 < \rho < r_{\mathrm{loc}}$ with $\alpha\rho > 2\delta$, and define

$$A(\rho) := \frac{2M}{(r_{\mathrm{loc}}/\rho)^{d_0/k_\star} - 1}.$$

If $\lambda(\rho)^2 \leq A(\rho)^2$, then (13) is immediate, since its right-hand side is zero. Assume therefore that $\lambda(\rho)^2 > A(\rho)^2$. If, toward a contradiction, (13) failed, then we would have

$$4\varepsilon^2 < \lambda(\rho)^2 - A(\rho)^2.$$

In particular, $\lambda(\rho) > 2\varepsilon$, so part (b) applies and yields

$$\left(\frac{r_{\mathrm{loc}}}{\rho}\right)^{d_0/k_\star} \leq 1 + \frac{2M}{\sqrt{\lambda(\rho)^2 - 4\varepsilon^2}}.$$

Rearranging,

$$\sqrt{\lambda(\rho)^2 - 4\varepsilon^2} \leq \frac{2M}{(r_{\mathrm{loc}}/\rho)^{d_0/k_\star} - 1} = A(\rho),$$

which contradicts $4\varepsilon^2 < \lambda(\rho)^2 - A(\rho)^2$. Hence $4\varepsilon^2 \geq \lambda(\rho)^2 - A(\rho)^2$, i.e.,

$$\varepsilon \geq \frac{1}{2}\big[\lambda(\rho)^2 - A(\rho)^2\big]_+^{1/2},$$

which is (13). This proves part (c). □

**From Theorem 4 to the informal Theorem 2.** The informal bound stated in Sec. 4.3 is the special case of part (c) obtained by fixing the inner radius to a constant fraction of the outer radius and discarding the squares.

**Corollary 1** (Distance-to-task-subspace lower bound; formal form of Theorem 2)**.** *Under the hypotheses of Theorem 4, choose* $\rho := r_{\mathrm{loc}}/2$ *and assume* $\alpha\rho > 2\delta$*, where* $\alpha = \sigma_0/2$ *is the local co-Lipschitz constant from Theorem 4. Then*

$$\sup_{z\in\mathcal{U}} \operatorname{dist}\big(\boldsymbol{\mu}_\star(z), \mathcal{S}\big) \;\geq\; \frac{c-\delta}{L} \;-\; \frac{C}{2^{d_0/k_\star}-1}, \tag{15}$$

*with* $c := \alpha r_{\mathrm{loc}}/4$ *and* $C := M$*.*

*Proof.* With $\rho = r_{\mathrm{loc}}/2$ one has $(r_{\mathrm{loc}}/\rho)^{d_0/k_\star} = 2^{d_0/k_\star}$, and part (c) of Theorem 4 reduces to

$$\varepsilon \;\geq\; \frac{1}{2}\Big[\,A^2 - B^2\,\Big]_+^{1/2}, \qquad A := \frac{\alpha r_{\mathrm{loc}}/2 - 2\delta}{L}, \quad B := \frac{2M}{2^{d_0/k_\star}-1}.$$

Since $\sqrt{A^2-B^2} \geq A - B$ whenever $A \geq B \geq 0$ (as $A + B \geq A - B$), we obtain

$$\varepsilon \;\geq\; \tfrac{1}{2}[A-B]_+ \;\geq\; \frac{\alpha r_{\mathrm{loc}}/4 - \delta}{L} - \frac{M}{2^{d_0/k_\star}-1}.$$

Identifying the right-hand side with $\frac{c-\delta}{L} - \frac{C}{2^{d_0/k_\star}-1}$ yields (15). □

Corollary 1 thus instantiates the informal Theorem 2 of the main text: the local constants $c$ and $C$ depend on the local co-Lipschitz constant $\alpha$, the confinement radius $r_{\mathrm{loc}}$, and the local diameter $M$, but not on the latent dimensions $d_0$ or $k_\star$. The decay in $2^{d_0/k_\star}$ is therefore the genuine source of the obstruction: when $d_0 \gg k_\star$, the second term in (15) is negligible, leaving a distance-to-$\mathcal{S}$ lower bound of order $(c-\delta)/L$. Theorem 4 is strictly stronger, since it provides a packing inequality that holds at every scale $\rho < r_{\mathrm{loc}}$ and quantifies the interaction between the prediction error $\delta$, the decoder Lipschitz constant $L$, and the confinement radius $\varepsilon$.

# C Empirical Validation of Task-Vector Properties

This section provides the computational and empirical details behind the evaluation of the task-vector properties `P0`–`P3`.

- Sec. C.1 describes the task-vector extraction procedure used throughout our analysis.
- Sec. C.2 reports the empirical evidence for the long-context properties `P0`–`P1`, complemented by an OLS probe decomposition for experiment `E3` in Sec. C.3.
- Sec. C.4 provides the computational and empirical details for the finite-context properties `P2`–`P3`.
- Sec. C.5 details the simplex-intervention experiment of Sec. 5.2.

## C.1 Extracting and Estimating Task Vectors

This subsection records the precise procedure used to extract residual-stream hidden states from the trained transformers and to estimate the task vectors $\hat{\boldsymbol{\theta}}_k$ that play the role of simplex anchors throughout Sec. 5. The extraction recipe and averaging-based estimator below are reused without modification by every empirical analysis in Sec. C (variance and additivity tests, simplex-projection coefficients $\beta_{t,k}$, simplex interventions) and by the orthogonal-subspace interventions in Sec. D. We write $\boldsymbol{H}^{(\ell)} = [\boldsymbol{h}_1^{(\ell)}, \ldots, \boldsymbol{h}_T^{(\ell)}]^\top$ for the collection of residual stream representations across positions. We consider a Pre-LayerNorm (LN) decoder-only transformer, where each layer updates a residual stream $\boldsymbol{H}^{(\ell-1)} \in \mathbb{R}^{T\times d}$ via

$$\widetilde{\boldsymbol{H}}^{(\ell)} = \boldsymbol{H}^{(\ell-1)} + \texttt{MHA}\big(\texttt{LN}(\boldsymbol{H}^{(\ell-1)})\big),$$
$$\boldsymbol{H}^{(\ell)} = \widetilde{\boldsymbol{H}}^{(\ell)} + \texttt{MLP}\big(\texttt{LN}(\widetilde{\boldsymbol{H}}^{(\ell)})\big).$$

We extract representations from the residual stream after the full layer update, i.e., from $\boldsymbol{H}^{(\ell)}$, and for each position $t$ collect the layer-wise trajectory $\{\boldsymbol{h}_t^{(\ell)}\}_{\ell=0}^{L-1}$.

**Extraction positions.** For discrete-token tasks (E1, E3, E4), the input sequence is $[s_1, \ldots, s_T]$ and we extract at all positions, $t \in \{1, \ldots, T\}$. For the linear regression task (E2), the sequence alternates between inputs and targets, $[\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2, \ldots, \boldsymbol{x}_{T/2}, y_{T/2}]$; we extract only at input positions $\boldsymbol{x}_t$, which are the prediction locations for $y_t$.

Having specified the residual-stream extraction recipe above, we now describe how the task-vector estimates $\hat{\boldsymbol{\theta}}_k$ themselves are computed from the extracted hidden states. This averaging-based estimator produces every $\hat{\boldsymbol{\theta}}_k$ used in the empirical evaluations of Sec. C.2–Sec. C.4.3 and in the simplex-/orthogonal-subspace interventions of Sec. C.5 and Sec. D.8; it is faithful to Definition 1 in the long-context limit.

**Estimation by averaging.** We estimate task vectors directly from the definition in Eq. 2. For each layer $\ell$ and a window of late context positions $\mathcal{T}_{\text{est}}$, we pool hidden states across positions and sequences for each task $k$:

$$\hat{\boldsymbol{\theta}}_k = \underbrace{\frac{1}{|\mathcal{T}_{\text{est}}|\, B} \sum_{t \in \mathcal{T}_{\text{est}}} \sum_{i:\, z_i = k} \boldsymbol{h}_{t,i}}_{\text{task-conditional mean}} - \underbrace{\frac{1}{K} \sum_{k'=1}^{K} \Big( \frac{1}{|\mathcal{T}_{\text{est}}|\, B} \sum_{t \in \mathcal{T}_{\text{est}}} \sum_{i:\, z_i = k'} \boldsymbol{h}_{t,i} \Big)}_{\text{grand mean}},$$

where the sums over $i$ run over all $B$ sequences assigned to each task. By construction $\sum_k \hat{\boldsymbol{\theta}}_k = \mathbf{0}$. $\mathcal{T}_{\text{est}}$ is a contiguous window of $\approx$ 15–30 *late* context positions, chosen so that the Bayesian posterior $\mathbb{P}(z \mid s_{\leq t})$ has largely concentrated and $\mathbb{E}[\boldsymbol{h}_t \mid z = k]$ approximates its long-context limit (Eq. 2); empirically, this is the regime where P0–P1 hold to high accuracy (Sec. C.2, Figure 9).

### C.2 Long-context stability and decoupling P0-P1

This subsection provides the empirical details behind the long-context properties P0–P1: we report the residual variance ratio across context positions (Sec. C.2.1), showing that task identity and the current token account for most of the variance in hidden states, and we test whether the conditional mean decomposes additively via ANOVA / ANCOVA (Sec. C.2.2).

#### C.2.1 Residual variance ratio

We measure long-context stability (P0) and decoupling (P1) jointly through the residual variance ratio $\widehat{\mathrm{Var}}(\boldsymbol{h}_t \mid z, s_t)/\widehat{\mathrm{Var}}(\boldsymbol{h}_t)$, the fraction of hidden-state variance that is *not* explained by the task identity $z$ together with the current token $s_t$. A small ratio at large $t$ indicates that the task and current token jointly account for almost all variation in $\boldsymbol{h}_t$, which is the empirical content of P0–P1. This complements Table 1(a) in the main text (which only reports the value at the last position) by displaying the full layer-by-layer trajectory across context positions for E1–E3.

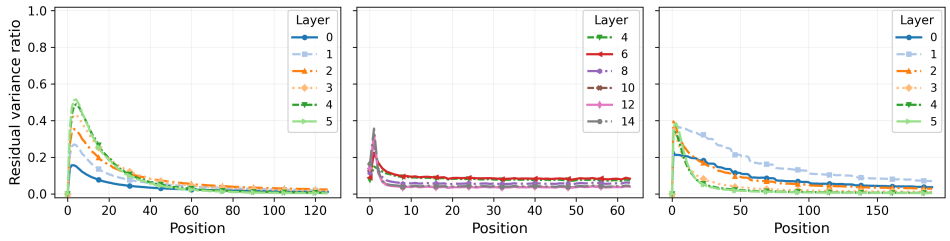



Figure 9: **Latent task and last token explain most variance in hidden states at large $t$.** Residual variance ratio $\mathrm{SS}_{\text{within}}/\mathrm{SS}_{\text{total}} = \widehat{\mathrm{Var}}(\boldsymbol{h}_t \mid z, s_t)/\widehat{\mathrm{Var}}(\boldsymbol{h}_t)$ as a function of context position for E1 (left), E2 (middle), and E3 (right). Each curve corresponds to one layer (post-MLP). The ratio decreases as context grows (the model gradually infers the latent) and is smaller in later layers (depth aids task inference).

Figure 9 reports the full residual-ratio curves across context positions; Table 1(a) (main text) summarizes the values at the last position.

### C.2.2 Testing additive separability

We test whether the conditional mean $\hat{\boldsymbol{\mu}}_{t,k,a} := \hat{\mathbb{E}}[\boldsymbol{h}_t \mid z = k, s_t = a]$ decomposes additively as $\hat{\boldsymbol{\mu}}_{t,k,a} \approx \hat{\boldsymbol{\mu}}_t + \hat{\boldsymbol{\theta}}_k + \hat{\boldsymbol{\nu}}_a$, where the task effect $\hat{\boldsymbol{\theta}}_k$ and token effect $\hat{\boldsymbol{\nu}}_a$ do not interact. The key idea is to compare an *additive* model against a *full* model that includes a task–token interaction term.

Fix a layer $\ell$ and position $t$. For experiments `E1` and `E3` with discrete tokens $a \in \mathcal{V}$, the additive model is

$$\boldsymbol{h}_t \;=\; \boldsymbol{\mu}_t + \boldsymbol{\theta}_k + \boldsymbol{\nu}_a,$$

while the full model allows a separate mean for each $(k, a)$ cell. This is a standard two-way ANOVA. A two-way decomposition of the between-cell variance gives

$$\underbrace{\sum_{k,a} \|\hat{\boldsymbol{\mu}}_{t,k,a} - \hat{\boldsymbol{\mu}}_t\|^2}_{\mathrm{SS}_{\text{between}}} \;=\; \underbrace{V \sum_{k} \|\hat{\boldsymbol{\theta}}_k\|^2}_{\mathrm{SS}_{\text{task}}} \;+\; \underbrace{K \sum_{a} \|\hat{\boldsymbol{\nu}}_a\|^2}_{\mathrm{SS}_{\text{token}}} \;+\; \underbrace{\sum_{k,a} \|\hat{\boldsymbol{\mu}}_{t,k,a} - \hat{\boldsymbol{\mu}}_t - \hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\nu}}_a\|^2}_{\mathrm{SS}_{\text{interaction}}},$$

where $\hat{\boldsymbol{\theta}}_k = (1/V) \sum_a \hat{\boldsymbol{\mu}}_{t,k,a} - \hat{\boldsymbol{\mu}}_t$ and $\hat{\boldsymbol{\nu}}_a = (1/K) \sum_k \hat{\boldsymbol{\mu}}_{t,k,a} - \hat{\boldsymbol{\mu}}_t$.

For experiment `E2`, where the current token $\boldsymbol{x}_t \in \mathbb{R}^D$ is a continuous covariate, the additive model is

$$\boldsymbol{h}_t \;=\; \boldsymbol{\mu}_t + \boldsymbol{\theta}_k + \boldsymbol{B}\,\boldsymbol{x}_t,$$

where $\boldsymbol{B}$ is a shared slope matrix, while the full model replaces $\boldsymbol{B}$ with task-specific slopes $\boldsymbol{B}_k$. Both models are fit by ordinary least squares (ANCOVA).

In both cases, we measure the *interaction proportion*: the fraction of explained variance attributable to the interaction term. For discrete tokens this is $\eta^2_{\text{interaction}} = \mathrm{SS}_{\text{interaction}}/\mathrm{SS}_{\text{between}}$. For continuous covariates, we define the analogous quantity $\eta^2_{\text{interaction}} = (R^2_{\text{full}} - R^2_{\text{additive}})/R^2_{\text{full}}$, i.e. the fraction of the full model's explanatory power that is due to task-specific slopes. A small $\eta^2_{\text{interaction}}$ confirms that the additive model is accurate.

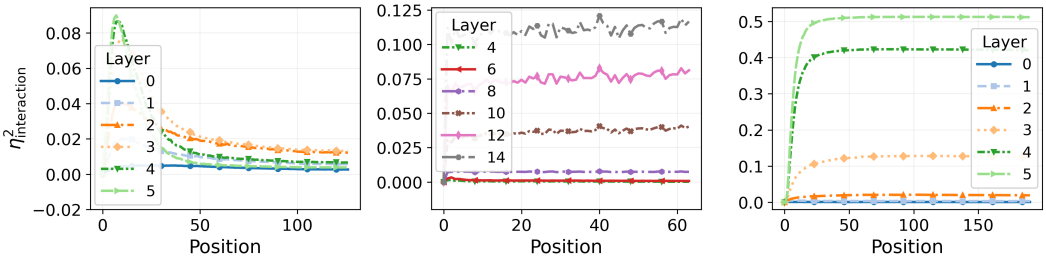



Figure 10: **Additive separability of task and token effects.** All three panels show the interaction proportion $\eta^2_{\text{interaction}}$: the fraction of explained variance due to the task–token interaction term. Left (`E1`): two-way ANOVA on discrete tokens. Middle (`E2`): ANCOVA on continuous covariates, $\eta^2_{\text{interaction}} = (R^2_{\text{full}} - R^2_{\text{additive}})/R^2_{\text{full}}$. Right (`E3`): two-way ANOVA on discrete tokens. Small values confirm that the additive model is accurate. Additivity holds broadly, with significant departures in late layers of `E2` and `E3`.

## C.3 OLS probe decomposition for experiment `E3`

This subsection complements the aggregate analysis in Sec. 5.1 with a layerwise variance decomposition of the hidden states in `E3`. The simplex-projected interpolation $R^2$ (Figure 2) plateaus at $0.60$ and $0.52$ in the last two layers, and the ANOVA interaction proportion (Table 1(b)) flags a parallel breakdown of additivity, but neither metric identifies what the residual interaction encodes. We regress each hidden state on three factors (the task label, the current token, and the model's own output logits) and use the logits as a diagnostic for residual-stream directions aligned with the next-token distribution. Including the logits absorbs essentially all of the late-layer interaction, indicating that the apparent breakdown of additivity is *not a genuine task–token interaction* but reflects hidden-state components devoted to computing the next-token prediction. The decomposition further shows that task identity is most cleanly represented in middle layers and is progressively subsumed into the next-token signal in later layers, providing a quantitative account of why task vectors are best extracted from middle layers [20].

**Probe construction.** We regress the residual-stream representation $\boldsymbol{h}_t$ on three feature groups computed from the same forward pass: the task label $z$, the current token $s_t$, and the model's output logit $\hat{\boldsymbol{\ell}}_t$. The first two are the variables that property P1 predicts to enter $\boldsymbol{h}_t$ additively, so any contribution to the late-layer interaction must be absorbed by the third. We use $\hat{\boldsymbol{\ell}}_t$ because it is precisely the direction of the residual stream that the unembedding reads off, and therefore serves as a known summary of the next-token-prediction signal carried by $\boldsymbol{h}_t$. Before fitting we subtract the per-position mean across sequences, $\bar{\boldsymbol{h}}_t := \boldsymbol{h}_t - \boldsymbol{h}_{\cdot,t}$, to suppress position-dependent variation induced by the rotary positional code. The OLS model is

$$\bar{\boldsymbol{h}}_t \;=\; \boldsymbol{W}_z\,\tilde{\boldsymbol{e}}_z \;+\; \boldsymbol{W}_s\,\tilde{\boldsymbol{e}}_{s_t} \;+\; \boldsymbol{W}_{\text{logit}}\,\hat{\boldsymbol{\ell}}_t \;+\; \boldsymbol{b} \;+\; \boldsymbol{\varepsilon}_t, \tag{16}$$

where $\tilde{\boldsymbol{e}}_z \in \{0,1\}^{K-1}$ and $\tilde{\boldsymbol{e}}_{s_t} \in \{0,1\}^{V-1}$ are the one-hot encodings of $z \in [K]$ and $s_t$ with the last column dropped (to avoid collinearity with the intercept), $\hat{\boldsymbol{\ell}}_t \in \mathbb{R}^V$ is the output logit vector, and $\boldsymbol{b} \in \mathbb{R}^d$ is an intercept. We fit on $2^{13}$ sequences from the fully trained model for experiment E3 ($K$=3 major tasks, no minor tasks), evaluated at positions $t \in [170, 190)$.

**Marginal and partial $R^2$.** For a feature group $A$, the *marginal* $R^2$ is the $R^2$ from regressing $\bar{\boldsymbol{h}}_t$ on $A$ alone (with intercept), and the *partial* $R^2$

$$\Delta R^2_A \;=\; \frac{R^2_{\text{joint}} - R^2_{-A}}{1 - R^2_{-A}} \tag{17}$$

quantifies the unique contribution of $A$, where $R^2_{-A}$ is the $R^2$ of the reduced model obtained from Eq. 16 by dropping the regressors in $A$ while keeping the intercept and the other two groups. Reporting both is necessary because the regressors overlap: $\hat{\boldsymbol{\ell}}_t$ is a deterministic linear function of the deepest hidden state, so it shares variance with both the task label ($R^2_{\text{task}\leftrightarrow\text{logit}}$=0.36) and the current token ($R^2_{\text{token}\leftrightarrow\text{logit}}$=0.35); the task and token indicators are nearly orthogonal ($R^2_{\text{task}\leftrightarrow\text{token}}$=0.028). The partial $R^2$'s for task and logits should therefore be read as contributions conditional on the linear span of the other groups.

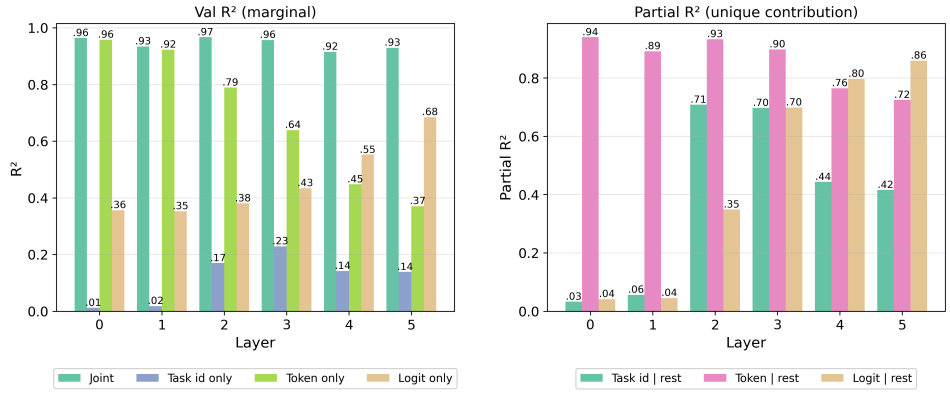



Figure 11: **OLS probe $R^2$ decomposition for experiment E3.** *Left*: marginal $R^2$ of each feature group alone. *Right*: partial $R^2$ (Eq. 17) after controlling for the other two groups. Results are shown layer-by-layer (post-MLP, layers 0–5) at context positions $t \in [170, 190)$, with per-position mean subtraction applied to $\boldsymbol{h}_t$ before fitting.

**Layerwise findings.** The decomposition pinpoints the source of the late-layer non-additivity: the next-token-prediction signal $\hat{\boldsymbol{\ell}}_t$ *progressively dominates the residual stream at deeper layers*, absorbing both task and token information and thereby driving the apparent breakdown of additivity flagged by Table 1(b) and Figure 2. As a corollary, the task label is encoded as a near-independent direction only at middle layers (2–3), while the current token retains a dominant unique contribution at all depths. Quantitatively, the joint model attains $R^2_{\text{joint}} \in [0.92, 0.97]$ at every layer, so the three groups together give a near-complete linear account of $\bar{\boldsymbol{h}}_t$. The next-token-prediction signal $\hat{\boldsymbol{\ell}}_t$ rises monotonically with depth: $\Delta R^2 \approx 0.04$–$0.05$ at layers 0–1, jumping to $0.36$ at layer 2 and reaching $0.86$ at layer 5. The task label has negligible unique contribution at layers 0–1 ($\Delta R^2 \approx 0.03$–$0.05$),

peaks at layers 2–3 ($\Delta R^2 \approx 0.70$–$0.71$), and decays to $\sim 0.42$ by layer 5. The current token carries the largest unique contribution at all depths (partial $R^2 \approx 0.89$–$0.94$ through layer 3, declining to $\sim 0.73$ at layer 5), consistent with property P1. The mechanism is now transparent: as $\boldsymbol{h}_t$ approaches the unembedding, $\hat{\boldsymbol{\ell}}_t$ absorbs information about both $z$ and $s_t$, the task label becomes redundant given $\hat{\boldsymbol{\ell}}_t$, and the additive interpolation model of Eq. 4 (which has no access to $\hat{\boldsymbol{\ell}}_t$) loses fitting capacity, which both explains the late-layer non-additivity and identifies layers 2–3 as the natural depth for task-vector extraction [20].

### C.4 Evaluating Finite-Context Properties (P2–P3)

This subsection provides the computational and empirical details behind the finite-context properties P2–P3: closed-form expressions for the oracle Bayesian posterior $\alpha_{t,k}$ in each experimental setting (Sec. C.4.1), the extraction of the simplex-constrained projection coefficients $\beta_{t,k}$ from hidden states (Sec. C.4.2), and additional posterior-alignment results across layers and with unconstrained coefficients (Sec. C.4.3).

#### C.4.1 Bayesian Posterior Computation under M1

This subsubsection derives the closed-form expressions for the oracle Bayesian posterior $\alpha_{t,k} := \mathbb{P}(z = k \mid s_1, \ldots, s_t)$ that serve as the comparison target throughout the paper. Under the task-retrieval mode M1 (Sec. 4.2) we take the prior to be uniform over the finite training support $\mathcal{Z}_{\text{train}} = [K]$, so that the integral in Eq. 6 reduces to a finite sum that admits a closed form for each of E1–E3. The resulting $\alpha_{t,k}$ play two roles in our analyses: they are the targets against which the empirical simplex coefficients $\beta_{t,k}$ are compared to assess Property P3 (Sec. 5.1, Figure 3; Sec. C.4.3), and they define the M1 predictive distribution used in the KL phase-transition analysis of Sec. 6. Throughout, we use the task index convention to refer to the latent variables for simplicity.

**Biased Dice (E1)** Recall that each task $z = k$ is parameterized by a probability vector $\boldsymbol{p}_k = (p_{k,1}, \ldots, p_{k,V}) \in \Delta^{V-1}$ over $V$ outcomes (the "faces of the die"). Tokens $s_1, s_2, \ldots$ are drawn i.i.d. from $\text{Categorical}(\boldsymbol{p}_k)$. The posterior at position $t$ factorizes as

$$\mathbb{P}(z = k \mid s_{\le t}) \;=\; \frac{\mathbb{P}\left(s_{\le t} \mid z = k\right)}{\sum_{k=1}^{K} \mathbb{P}\left(s_{\le t} \mid z = k\right)} = \frac{\prod_{\tau=1}^{t} p_{k,s_\tau}}{\sum_{k'=1}^{K} \prod_{\tau=1}^{t} p_{k',s_\tau}}.$$

Equivalently, let

$$n_a(t) := \sum_{\tau=1}^{t} \mathbf{1}\{s_\tau = a\} \tag{18}$$

be the count of token $a$ in the first $t$ tokens, the log-likelihood is

$$\log \mathbb{P}(s_{\le t} \mid z = k) \;=\; \sum_{a=1}^{V} n_a(t)\, \log p_{k,a}.$$

The posterior is obtained by normalizing across all $K$ tasks:

$$\alpha_{t,k} \;=\; \frac{\prod_{a=1}^{V} p_{k,a}^{\,n_a(t)}}{\sum_{k'=1}^{K} \prod_{a=1}^{V} p_{k',a}^{\,n_a(t)}}. \tag{19}$$

**Noisy Linear Regression (E2)** Recall that each task $z = k$ is parameterized by a weight vector $\boldsymbol{w}_k \in \mathbb{R}^D$. The sequence consists of input–output pairs $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots$ where $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$ and $y_t = \boldsymbol{w}_k^\top \boldsymbol{x}_t + \varepsilon_t$ with $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ independently.

At position $t$, define $\boldsymbol{X}_{\le t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)^\top \in \mathbb{R}^{t \times D}$ and $\boldsymbol{y}_{\le t} = (y_1, \ldots, y_t)^\top \in \mathbb{R}^t$. The cumulative log-likelihood under task $k$ is

$$\log p(\boldsymbol{y}_{\le t} \mid \boldsymbol{X}_{\le t}, z = k) \;=\; -\frac{t}{2} \log(2\pi\sigma^2) \;-\; \frac{1}{2\sigma^2} \sum_{\tau=1}^{t} (y_\tau - \boldsymbol{w}_k^\top \boldsymbol{x}_\tau)^2.$$

We define the cumulative sum of squared residuals $\mathrm{SSR}_k(t) = \sum_{\tau=1}^{t}(y_\tau - \boldsymbol{w}_k^\top \boldsymbol{x}_\tau)^2$. Because the filtering posterior at position $t$ conditions only on past observations $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})$, the SSR is evaluated at $t-1$:

$$\alpha_{t,k} \;=\; \frac{\exp\!\big(-\frac{1}{2\sigma^2}\,\mathrm{SSR}_k(t-1)\big)}{\sum_{k'=1}^{K}\exp\!\big(-\frac{1}{2\sigma^2}\,\mathrm{SSR}_{k'}(t-1)\big)}, \tag{20}$$

where $\mathrm{SSR}_k(0) = 0$ (no observations available at the first position).

**Latent Markov Chain (E3)** Recall that each task $z = k$ is parameterized by a transition kernel $\boldsymbol{P}^{(k)}$. The state space is $\mathcal{V}$. The first tokens carry no task-discriminative information; therefore $\alpha_{t,k} = 1/K$ for $t = 1$. For $t > 1$, each new token $s_t$ provides a likelihood contribution:

$$\mathbb{P}(s_t = a \mid s_{<t},\, z = k) \;=\; \boldsymbol{P}^{(k)}_{s_{t-1},a},$$

The posterior accumulates these contributions:

$$\mathbb{P}(z = k \mid s_{\le t}) \;=\; \frac{\mathbb{P}\left(s_{\le t} \mid z = k\right)}{\sum_{k'=1}^{K}\mathbb{P}\left(s_{\le t} \mid z = k'\right)}$$

which yields

$$\alpha_{t,k} \;=\; \frac{\prod_{\tau=2}^{t}\boldsymbol{P}^{(k)}_{s_{\tau-1},s_\tau}}{\sum_{k'=1}^{K}\prod_{\tau=2}^{t}\boldsymbol{P}^{(k')}_{s_{\tau-1},s_\tau}}. \tag{21}$$

### C.4.2 Computing projection coefficients $\beta_{t,k}$

This subsubsection details the two-step projection used to extract the simplex-constrained coefficients $\beta_{t,k}$ that appear in the interpolation model of Eq. 4 (Sec. 5). These coefficients are the empirical proxies for the Bayesian posterior weights $\alpha_{t,k}$ used in the posterior-alignment property P3, and the procedure below is the one used to produce every $\beta_{t,k}$ reported in Figure 3. Let $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_K \in \mathbb{R}^d$ be the centered task vectors (Sec. C.1), and let

$$\bar{\boldsymbol{h}}_t \;=\; \boldsymbol{h}_t^{(\ell)} - \hat{\boldsymbol{\mu}}_t$$

be the centered hidden state at position $t$ and layer $\ell$, where $\hat{\boldsymbol{\mu}}_t$ is the empirical grand mean. Following the main text, we obtain $\beta_{t,k}$ by projecting $\bar{\boldsymbol{h}}_t$ onto the task-vector simplex $\big\{\sum_k \beta_k\,\hat{\boldsymbol{\theta}}_k \,:\, \beta_k \ge 0,\, \sum_k \beta_k = 1\big\}$, implemented in two steps.

**Step 1: Affine projection.** We first find coefficients satisfying $\bar{\boldsymbol{h}}_t \approx \sum_{k=1}^{K}\beta_{t,k}\,\hat{\boldsymbol{\theta}}_k$ with $\sum_k \beta_{t,k} = 1$, without enforcing nonnegativity. Using $\hat{\boldsymbol{\theta}}_K$ as an anchor reduces this to an unconstrained least-squares problem in $K-1$ variables, solved via the pseudoinverse of the matrix of pairwise differences $(\hat{\boldsymbol{\theta}}_1 - \hat{\boldsymbol{\theta}}_K, \cdots, \hat{\boldsymbol{\theta}}_{K-1} - \hat{\boldsymbol{\theta}}_K)$. The resulting affine coefficients satisfy sum-to-one but may be negative.

**Step 2: Simplex projection.** To enforce $\beta_{t,k} \ge 0$, we project the affine coefficients onto $\Delta^{K-1}$ via the standard Euclidean simplex projection [13]: sort in decreasing order and apply a uniform threshold so that the result is nonnegative and sums to one. The projected coefficients $\beta_{t,k}$ are what Sec. 5 refers to as the simplex-projected interpolation coefficients.

### C.4.3 Additional posterior alignment results

Figure 3 in Sec. 5 reports the simplex-projected coefficients $\beta_{t,k}$ at a representative middle layer. Here we provide two complementary views: alignment at later layers, and unconstrained (pre-projection) coefficients. Together, these views demonstrate that posterior alignment (property P3) is *robust*, in the sense that: it holds across layers (not just the representative middle layer of the main text) and survives removal of the simplex projection (not just under the constrained estimator), confirming that the agreement between $\beta_{t,k}$ and $\alpha_{t,k}$ reflects a property of the hidden states themselves rather than an artifact of layer choice or the projection procedure.

**Later layers.** At later layers the projection coefficients track the Bayesian posterior $\alpha_{t,k}$ even more closely than at the middle layer shown in the main text.

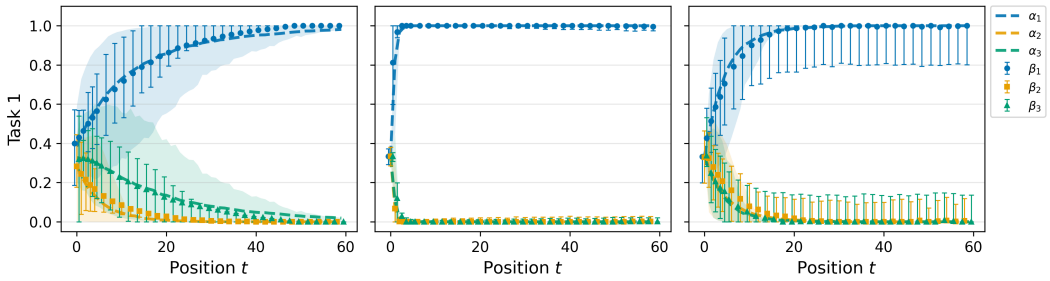



Figure 12: Posterior alignment at later layers: simplex-projected coefficients $\beta_{t,k}$ (markers with error bars) vs. Bayesian posterior $\alpha_{t,k}$ (dashed lines with shaded 10–90$^{\text{th}}$ percentile bands) for data generated from task 1. Compared to the middle-layer results in Figure 3, the agreement is tighter across all three tasks. Left: Dice (`E1`, layer 5). Middle: linear regression (`E2`, layer 15). Right: latent Markov (`E3`, layer 5).

**Affine (pre-projection) coefficients.** For completeness, we also report the affine coefficients $\beta^{\text{aff}}_{t,k}$, obtained by the same affine projection described in Sec. C.4.2 but without the simplex step (sum-to-one is enforced, but coefficients may be negative or exceed 1). The close match with the posterior even before simplex projection confirms that the alignment is not an artifact of the projection procedure.

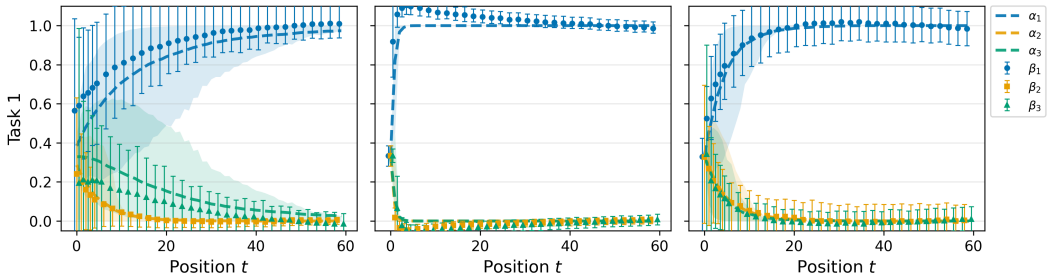



Figure 13: Affine projection coefficients $\beta^{\text{aff}}_{t,k}$ (markers with error bars) vs. Bayesian posterior $\alpha_{t,k}$ (dashed lines with shaded percentile bands). No simplex projection is applied; coefficients may be negative or exceed 1. Left: Dice (`E1`, layer 3). Middle: linear regression (`E2`, layer 9). Right: latent Markov (`E3`, layer 3).

**Summary.** Taken together, Figures 12 and 13 show that posterior alignment $\beta_{t,k} \approx \alpha_{t,k}$ is *not* contingent on the specific layer chosen for visualization in the main text, nor on the simplex-projection step in the estimator: the alignment persists at later layers (and in fact tightens) and survives when the nonnegativity step is removed (Figure 13 uses $\beta^{\text{aff}}_{t,k}$ in place of the simplex-projected $\beta_{t,k}$).

### C.5 Simplex Intervention Details

This subsection details the simplex-intervention experiment of Sec. 5.2: at a chosen layer $\ell$ we replace the task-subspace component of the residual stream with an arbitrary point $\sum_k \alpha^*_{t,k}\hat{\boldsymbol{\theta}}_k$ in the task-vector simplex (with $\boldsymbol{\alpha}^*_t$ drawn from $\Delta^{K-1}$), leave the orthogonal complement untouched, and measure how closely the steered model's output matches the Bayesian mixture prediction that $\boldsymbol{\alpha}^*_t$ dictates. The purpose is causal: the alignment between $\beta_{t,k}$ and $\alpha_{t,k}$ established by Property `P3` (Sec. 5.1) is observational, and could in principle reflect a hard task-selection rule rather than a true continuous interpolation. The simplex steering, paired with the *mode-output* baseline introduced below, addresses both points: it shows that the model's output tracks $\boldsymbol{\alpha}^*_t$ throughout the simplex (not merely near vertices), supporting the main-text claim that the task-vector subspace is the variable through which the model interpolates between training tasks. Let

$$\boldsymbol{P}_{\text{task}} = \hat{\boldsymbol{\Theta}}\,(\hat{\boldsymbol{\Theta}}^\top\hat{\boldsymbol{\Theta}})^\dagger\hat{\boldsymbol{\Theta}}^\top \tag{22}$$

denote the orthogonal projector onto the task subspace, where $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{d\times K}$ stacks the centered task vectors as columns and $\boldsymbol{A}^\dagger$ denotes the pseudoinverse of the matrix $\boldsymbol{A}$. We substitute the task-subspace component with a target interpolation, replacing the model's original coefficients $\beta_{t,k}$ with

coefficients $\alpha^*_{t,k}$ randomly drawn from the simplex $\Delta^{K-1}$:

$$\boldsymbol{h}_t^{(\ell),\mathrm{int}} = \hat{\boldsymbol{\mu}}_t + \sum_{k=1}^{K} \alpha^*_{t,k}\, \hat{\boldsymbol{\theta}}_k + (\boldsymbol{I} - \boldsymbol{P}_{\mathrm{task}})\big(\boldsymbol{h}_t^{(\ell)} - \hat{\boldsymbol{\mu}}_t\big).$$

This steers the hidden representation to a chosen point in the task-vector simplex while leaving the orthogonal complement unchanged. The model then completes the forward pass from layer $\ell$ onward using the modified hidden state.

**Comparing with an additional baseline: Mode-output.** A central claim of Sec. 5.2 is that task vectors implement a *continuous interpolation* over tasks rather than a discrete selection of the most likely one. The natural alternative hypothesis is a *hard-selection* (nearest-neighbor) mechanism: the model identifies the dominant task $k^* = \arg\max_k \alpha^*_{t,k}$ and predicts as if that task were certain. To rule this out, we compare simplex steering against a *mode-output* baseline that realizes this hard-selection rule using the oracle Bayesian posterior over tasks. Across all three experiments, simplex steering tracks the target mixture *uniformly across* $\Delta^{K-1}$, while the mode baseline matches the model *only near the vertices* and degrades sharply in the interior. This shows that task vectors cannot be reduced to a nearest-neighbor or hard-assignment mechanism, strengthening the main-text conclusion that they encode and combine task information continuously across the simplex.

**Mode-output baseline.** Given target coefficients $\boldsymbol{\alpha}^*_t$, the dominant task is

$$k^* \;=\; \arg\max_k \alpha^*_{t,k},$$

and the mode-output prediction is the Bayesian-optimal next-token distribution under that single task, $\mathbb{P}(\cdot \mid z = k^*, s_t)$. This baseline is computed entirely from the ground-truth generative model and does *not* involve a forward pass through the trained model; as such, it provides the strongest realization of the hard-selection alternative defined above and an upper bound on the performance any nearest-neighbor mechanism in task space could achieve.

**Empirical comparison.** Figure 14 reports the prediction error (KL divergence or RMSE, depending on the experiment) of simplex steering and the mode-output baseline against the target mixture across $\Delta^{K-1}$ for `E1`–`E3`. Simplex steering attains uniformly low error throughout the simplex, including the interior region where multiple tasks contribute comparably and any hard-selection rule must discard substantial posterior mass. The mode baseline, by contrast, matches the target only near the vertices and degrades sharply toward the interior, with error growing roughly with the entropy of $\boldsymbol{\alpha}^*_t$. This pattern is precisely what the interpolation hypothesis predicts and what the hard-selection hypothesis cannot accommodate.

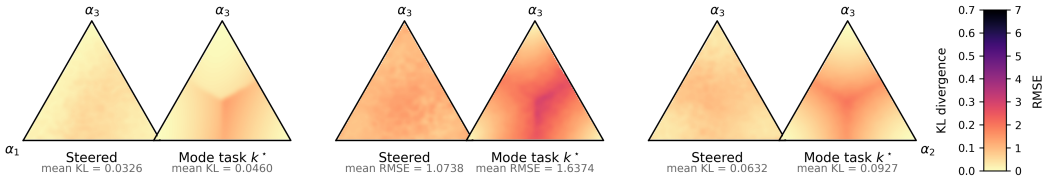



Figure 14: **Simplex steering outperforms the mode-output baseline across the entire task simplex.** Each panel pair shows error (KL divergence or RMSE) under simplex steering (*Steered*) and the mode-output baseline (*Mode task* $k^*$) for `E1` (left), `E2` (middle), and `E3` (right). Simplex steering achieves uniformly low error throughout the simplex interior, whereas the mode baseline (which selects the pure-task output for the dominant task, and is equivalent to an idealised hard-selection task-vector injection) performs well only near the vertices and degrades in the interior, where task mixtures are more balanced.

### C.6 Alternative task-vector estimators

The paper estimates task vectors by class means, replacing the expectation in Eq. 2 by an empirical average, as in Hendel et al. [20] and Todd et al. [51]. Under `P0` and `P1`, any estimator that recovers the task subspace estimates the same vectors $\{\boldsymbol{\theta}_k\}_{k\le K}$, so the results of the paper should not depend on this choice. We check this on the major-only `E3` model of Sec. 5 at layer 3, with the model trained

Table 5: **Task-vector estimators under the paper's evaluation pipeline (major-only E3 model, layer 3).** P2 interpolation $R^2$; Hellinger distance $H(\beta, \alpha)$ of Eq. 23; fraction of positions at which $\beta_t$ and $\alpha_t$ name the same task; held-out $R^2$ of $\alpha_t$ regressed on $\beta_t$; cluster purity for the unsupervised estimators. **Bottom:** simplex injection with the procedure of Fig. 4, mean KL divergence to the target mixture before and after steering.

| estimator | labels | P2 $R^2$ | $H(\beta, \alpha)$ | same task | $R^2(\alpha$ on $\beta)$ | purity |
|---|---|---|---|---|---|---|
| Class means (ours) | task | 0.90 | 0.11 | 1.00 | 0.98 | – |
| Diagonal LDA | task | 0.90 | 0.10 | 1.00 | 0.98 | – |
| Token-conditioned PCA, $k$-means | none | 0.90 | 0.09 | 1.00 | 0.98 | 1.00 |
| Pooled PCA, $k$-means | none | 0.81 | 0.10 | 1.00 | 0.98 | 1.00 |
| Random, matched norm | – | 0.58 | 0.61 | 0.72 | 0.16 | – |

*Simplex injection, mean KL divergence (before → after steering)*

| | | | |
|---|---|---|---|
| Class means (ours) | $0.231 \to 0.062$ | Pooled PCA, $k$-means | $0.233 \to 0.093$ |
| Token-conditioned PCA, $k$-means | $0.232 \to 0.063$ | Random, matched norm | $0.231 \to 0.223$ |

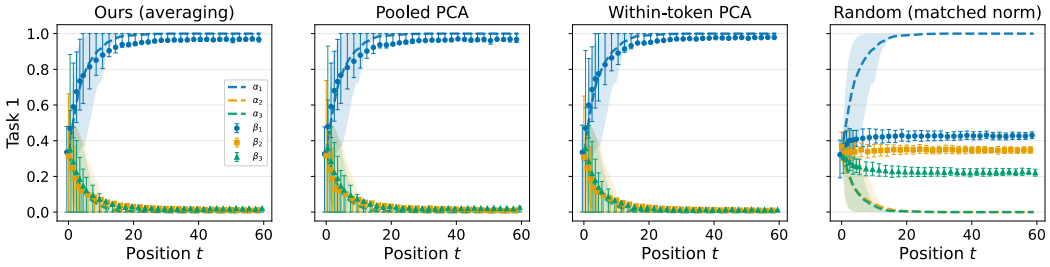



Figure 15: P3 **by estimator (major-only** E3 **model, layer** 3**, task** 1**).** Simplex coefficients $\beta_{t,k}$ (markers with percentile bars) against the oracle posterior $\alpha_{t,k}$ (dashed), as in Fig. 3, for class means, pooled PCA, token-conditioned PCA, and random vectors of matched norm. The three estimators that recover the task subspace track the posterior; random vectors do not.

at diversity exponent $m = 6$ as a second case. We run four alternatives through the same evaluation pipeline as the class means, with the same grand mean, token encodings and simplex solver. Diagonal linear discriminant analysis (LDA) rescales each coordinate of the class means by its within-class standard deviation and uses task labels. Token-conditioned PCA applies principal component analysis to hidden states after conditioning on the current token and then $k$-means, and uses no labels. Pooled PCA applies principal component analysis to all hidden states and then $k$-means, and uses no labels. Random vectors with the same norm as ours serve as a reference. Since P0 and P1 depend on the hidden states alone, the estimators can differ only on P2, P3 and the simplex injection of Sec. 5.2.

**Evaluation.** We measure P2 by the interpolation $R^2$ of Eq. 4. For P3 we compare the coefficients $\beta_t$ to the oracle posterior $\alpha_t$ on held-out positions in three ways. The first is the Hellinger distance

$$H(\beta_t, \alpha_t) \;=\; \Big(\tfrac{1}{2}\textstyle\sum_{k\le K}\big(\sqrt{\beta_{t,k}} - \sqrt{\alpha_{t,k}}\big)^2\Big)^{1/2} \in [0,1], \tag{23}$$

averaged over positions, where smaller is better. The second is the fraction of positions at which the largest coefficient and the largest posterior probability name the same task. The third is the $R^2$ of a regression of $\alpha_t$ on $\beta_t$, fitted on one half of the held-out data and evaluated on the other. The simplex injection follows Fig. 4. For the two unsupervised estimators, $k$-means recovers the $K$ vertices only up to a permutation of the task labels. We fix the permutation on the estimation split, by matching each cluster to the task it most often contains, and keep it fixed on the evaluation split, so labels enter only as a relabeling of the recovered vertices. In our runs every cluster contains sequences of a single task (purity 1.00), so the clustering alone determines the partition.

**Results.** Table 5 and Figs. 15–16 show that class means, diagonal LDA and token-conditioned PCA give the same P2, P3 and injection results, and that random vectors of matched norm give none of them. Diagonal LDA reproduces the class means because the within-class rescaling rotates each task vector only slightly (cosine at least 0.94) and changes the simplex coefficients by at most 0.012 on average. Token-conditioned PCA recovers the task vectors without any labels, which shows that the

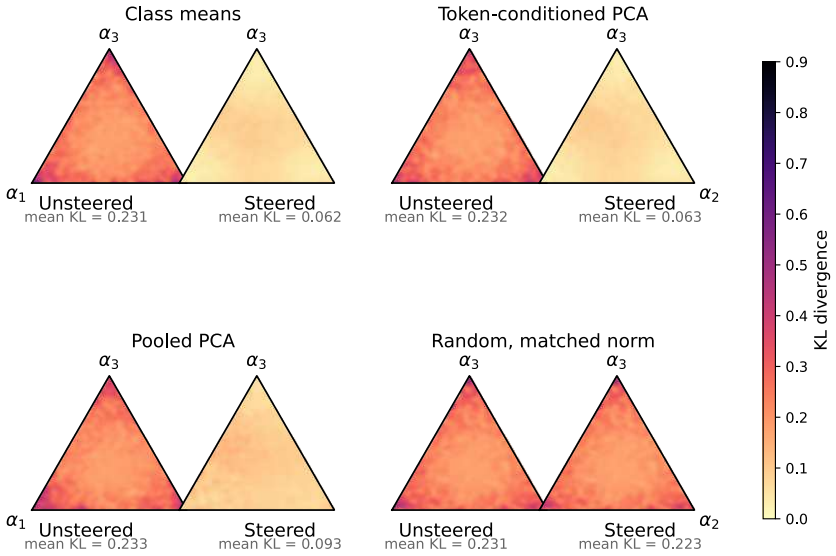



Figure 16: **Simplex injection by estimator** (procedure of Fig. 4, E3, layer 3). Each pair shows the KL divergence to the target mixture over the simplex of targets, before steering (left triangle) and after (right triangle), on a shared color scale, with the mean printed beneath; the means match Table 5.

geometry is a property of the hidden states and not of the supervision. Pooled PCA, the one estimator that does not condition on the current token, is the one that degrades. Token variance is comparable to task variance at this layer, so its leading components mix task and token directions, which lowers the P2 $R^2$ to $0.81$ and the steering quality to $0.093$. On the $m = 6$ model this degradation is larger (fraction of positions naming the same task $0.51$), whereas token-conditioned PCA stays at $0.99$ and class means at $1.00$. Finally, a linear probe regressing hidden states on one-hot task indicators is the class-mean estimator exactly, since the least-squares coefficient of each task is that task's mean hidden state.

Our goal is to validate the class-mean task vector widely used in prior work, not to advocate it over alternatives. Under P0 and P1, class means, diagonal LDA and token-conditioned PCA all estimate the same $\{\boldsymbol{\theta}_k\}_{k \le K}$, and the posterior-alignment and intervention results of the paper are not artifacts of one estimator.

### C.7 Causal identification of individual vertices

The injection of Sec. 5.2 sets the task-subspace component of $\boldsymbol{h}_t$ to $\sum_k \alpha_k^* \hat{\boldsymbol{\theta}}_k$ and scores the output against the Bayesian predictive of that specific mixture $\alpha^*$. Its success could in principle reflect only the span $\mathrm{col}(\hat{\boldsymbol{\Theta}})$, with any basis of the same plane steering equally well. To check this we repeat the injection of Fig. 4 on the major-only E3 model (layer 3, per-position injection, 10k Dirichlet targets, a new seed) with bases that span the identical task plane but assign the vertices differently: the frame rotated in-plane by $30°$ or $60°$, two vertices swapped, all three cyclically relabeled, and 12 random in-plane rotations $\varphi \sim U(0°, 360°)$. Every variant spans exactly the plane of our vectors; only the vertex assignment varies. The unsteered KL is $0.233$, and steering to the single most likely task $k^\star$ instead of the mixture gives $0.092$.

Table 6 and Fig. 17 show that steering error grows monotonically with vertex misalignment at fixed plane. A typical wrong-vertex basis in the same plane (random rotation, $0.143 \pm 0.055$) fails as badly as a swap or a cyclic relabeling, so the $30°$ and $60°$ rows understate the typical effect. Steering therefore requires the correct assignment of vertices to tasks, not just the span, and the intervention identifies the individual vertices. This agrees with P3, which is evaluated coordinate by coordinate ($\beta_k$ tracks $\alpha_k$ with $R^2 = 0.98$), and with Sec. C.6, where class means, diagonal LDA and token-conditioned PCA recover the same vertices. The first row reproduces the $0.231 \to 0.062$ of Fig. 4 with a new seed.

Table 6: **Same plane, different vertices.** Mean KL to the target mixture after steering through bases that share the task plane with our class-mean vertices but differ in vertex assignment; unsteered $0.233$.

| basis (identical task plane) | per-vertex cosine to ours | steered KL |
|---|---|---|
| Ours (class-mean vertices) | $1.00, 1.00, 1.00$ | $\mathbf{0.062}$ |
| Rotated $30^\circ$ in-plane | $0.87$ ($\times 3$) | $0.075$ |
| Rotated $60^\circ$ in-plane | $0.50$ ($\times 3$) | $0.112$ |
| Two vertices swapped | $-0.41, -0.41, 1.00$ | $0.143$ |
| Cyclically relabeled (all wrong) | $-0.41, -0.44, -0.64$ | $0.198$ |
| Random in-plane rotation (12 draws) | – | $0.143 \pm 0.055$ (range $0.067$–$0.215$) |




Figure 17: **Injection through same-plane bases.** KL divergence to the target mixture over the simplex of targets, before steering (left triangle of each pair) and after (right), on a shared color scale, with the mean printed beneath. With the plane held fixed, steering error grows with vertex misalignment: $0.062$, $0.112$, $0.198$.

# D Further Details about the Two Inference Modes

This appendix collects the empirical evidence supporting the two-mode picture introduced in Sec. 6, complementing the KL-based transition of Sec. 6, the orthogonality results of Sec. 6.1, and the causal interventions of Sec. 6.3. The material is organized as follows.

- Sec. D.1: detailed discussion on the training distribution used in experiments of Sec. 6.1.
- Sec. D.2: training dynamics of in-distribution and out-of-distribution loss, providing a performance-based view of the same transition between `M1` and `M2`.
- Sec. D.4: a complementary geometric view of the representation orthogonality, tracing simplex trajectories of hidden-state projections under low and high task diversity.
- Sec. D.5: pretrained-LLM (Qwen2.5-7B) experiments that extend the two-mode trajectory analysis from synthetic data to a real language model.
- Sec. D.8: full per-layer breakdowns of the two complementary causal interventions of Table 2, with implementation details (Sec. D.9) and a decomposition of the optimized orthogonal directions (Sec. D.10) showing they encode running context statistics.

## D.1 Major/Minor Training Mixture

This subsection describes the major/minor task-mixture training distribution used throughout Sec. D for the coexistence-of-modes experiments of Sec. 6. The minor-pool size $N_{\text{minor}}$ is the central experimental knob: small $N_{\text{minor}}$ allows every task to be memorized (favoring `M1`), while large $N_{\text{minor}}$ makes individual minor tasks unmemorizable and forces the model toward in-context generalization (favoring `M2`).

**Task pools.** Before training begins, two disjoint task pools are sampled once from the task-specific prior and held fixed throughout:

- **Major pool.** Three tasks $\{z_1^{\text{maj}}, z_2^{\text{maj}}, z_3^{\text{maj}}\}$ drawn from the prior $\pi$ (e.g. $\text{Dir}(\mathbf{1}_V)$ for `E1` and `E3`, $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$ for `E2`).
- **Minor pool.** $N_{\text{minor}}$ tasks $\{z_1^{\text{minor}}, \ldots, z_{N_{\text{minor}}}^{\text{minor}}\}$ drawn independently from the same prior $\pi$, conditioned on being disjoint from the major pool. We allow $N_{\text{minor}} = 0$ as a degenerate case in which the minor pool is empty and all sequences are drawn from the major pool only.

The minor-pool size $N_{\text{minor}}$ is the primary control variable. The full diversity sweep (Sec. A) covers $N_{\text{minor}} \in \{0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$; Figure 6 displays the five-value subset $N_{\text{minor}} \in \{0, 1, 4, 16, 1024\}$ in all three panels (`E1`, `E2`, `E3`), so that the legend values match exactly across experiments.

**Sequence sampling.** For $N_{\text{minor}} \geq 1$, each training sequence is generated by the following two-stage procedure:

1. **Pool selection.** With probability $p_{\text{minor}} = 0.1$, draw from the minor pool; with probability $1 - p_{\text{minor}} = 0.9$, draw from the major pool.
2. **Task selection.** Sample a task $z$ uniformly at random from the selected pool, then generate a sequence of length $T$ from $p_z$.

For the degenerate case $N_{\text{minor}} = 0$ the minor pool is empty: the Bernoulli pool-selection step is skipped and every sequence is drawn from the major pool (equivalently, the effective $p_{\text{minor}}$ collapses to 0). For $N_{\text{minor}} \geq 1$, this gives major tasks a dominant 9:1 weight over the minor pool regardless of $N_{\text{minor}}$, so each individual minor task receives weight $p_{\text{minor}}/N_{\text{minor}} = 0.1/N_{\text{minor}}$, vanishingly small for large $N_{\text{minor}}$.

**OOD tasks.** Each OOD task is obtained by drawing a fresh sample from the same task-specific prior used to construct the major and minor pools: a Dirichlet draw over categorical distributions for `E1` and `E3`, and an isotropic Gaussian draw over weight vectors for `E2`. These draws are made independently of the stored training pools and are not cached: every batch of OOD tasks is generated afresh from the prior. Because the prior is continuous, with probability one the resulting task does not coincide with any task in the major or minor pool, so OOD tasks are never seen during training. This ensures that OOD performance reflects genuine out-of-distribution generalization rather than minor-task memorization.

**Relation to task diversity.** When $N_{\text{minor}} \in \{0, 1\}$ the minor pool is empty or contains a single task, which the model can readily memorize; for small $N_{\text{minor}}$ the model essentially extends task-retrieval behavior to the minor pool. As $N_{\text{minor}}$ grows, the per-task training weight $0.1/N_{\text{minor}}$ becomes negligible and no individual minor task can be memorized, forcing the model to develop a context-based generalization strategy. The transition in representation geometry visible in Figure 6 reflects this shift.

### D.2 In-distribution and out-of-distribution loss

To complement the KL-based comparison in Sec. 6, we also examine the training dynamics of in-distribution (ID) and out-of-distribution (OOD) loss. ID sequences are drawn from the major/minor training mixture and OOD sequences from fresh prior draws disjoint from both training pools; see Sec. D.1 for the precise sampling procedure. These plots provide a direct performance-based view of the same `M1`-to-`M2` transition: as task diversity increases, the model enters a regime in which it obtains strong OOD performance even though its predictions are no longer best described by `M1` (the exact Bayesian posterior over the memorized task set).

Figure 18 reports ID and OOD loss curves for `E1`, `E2` and `E3`. The main qualitative pattern is that OOD loss remains low, or improves substantially during training, in the same region where the KL comparison favors `M2`. This supports the interpretation that `M2` is not merely a worse fit to the `M1` baseline but a distinct inference mode that supports effective OOD prediction.

### D.3 Capacity versus per-task data

In the diversity sweep of Fig. 5, the minor tasks share a fixed prior mass $p_{\text{minor}} = 0.1$ (Sec. D.1). Increasing the number of minor tasks $N_{\text{minor}} = 2^m$ therefore also shrinks the training data that each minor task receives, to a share of $0.1/2^m$. The transition from `M1` to `M2` along $m$ may thus be driven by the data available per task, by limited model capacity, or by both. This section separates the two with two sweeps on `E3` (30k steps, as in Sec. E). The first varies the hidden dimension $d$ at fixed $p_{\text{minor}}$. The second varies $m$ while holding each minor task's share of the training data fixed. We use the mode diagnostic $\log(\text{KL}_{\text{Bayesian}}/\text{KL}_{\text{extrapolative}})$ of Sec. 6, negative for `M1`, and the transition point $m^*$, the value of $m$ at which the diagnostic changes sign.

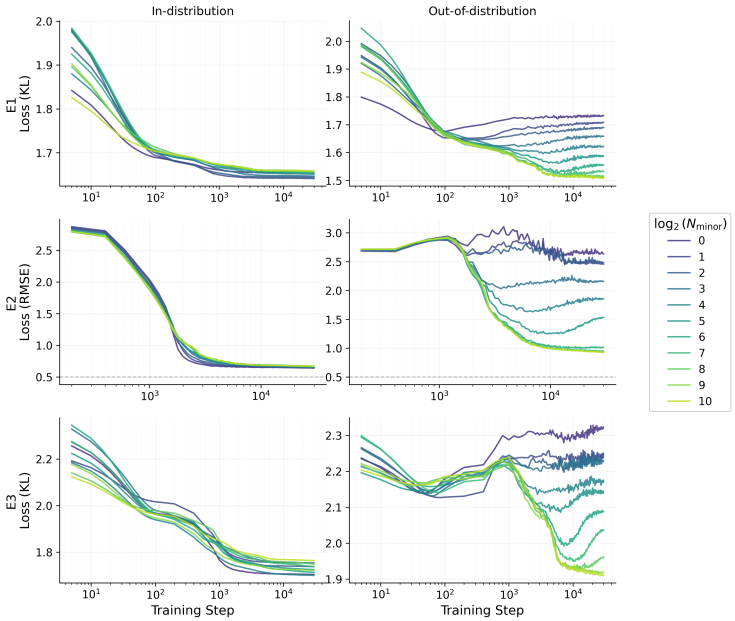



Figure 18: **Training dynamics of in-distribution and out-of-distribution loss.** Panels (top to bottom): `E1`, `E2`, `E3`. Each panel compares ID and OOD performance over training. Strong OOD performance in the later training regime of high task diversity is consistent with the emergence of the extrapolative inference mode `M2` (cf. Sec. 6).

Table 7: **Capacity and per-task data. Top:** mode diagnostic versus hidden dimension $d$ at fixed $p_{\text{minor}} = 0.1$; $d = 128$ is the paper's architecture. **Bottom:** the sweep at $d = 128$ in which $p_{\text{minor}} \propto N_{\text{minor}}$, so that every minor task keeps the share of training data it has at $m = 6$, where the two sweeps coincide (total data and steps unchanged).

| width $d$ | $m$=2 | $m$=4 | $m$=6 | $m$=8 | $m$=10 | transition $m^*$ |
|---|---|---|---|---|---|---|
| 32 | $-2.77$ | $-0.88$ | $+0.84$ | $+2.02$ | $+1.99$ | $\approx 5.0$ |
| 64 | $-3.43$ | $-2.42$ | $-0.45$ | $+2.09$ | $+2.46$ | $\approx 6.4$ |
| 128 | $-4.15$ | $-2.99$ | $-1.45$ | $+0.90$ | $+2.99$ | $\approx 7.2$ |
| 256 | $-4.67$ | $-3.65$ | $-2.41$ | $-0.48$ | $+2.81$ | $\approx 8.3$ |
| 512 | $-4.40$ | $-3.99$ | $-2.80$ | $-1.16$ | $+2.77$ | $\approx 8.6$ |
| sweep ($d = 128$) | $m$=2 | $m$=4 | $m$=6 | $m$=8 | | |
| fixed $p_{\text{minor}} = 0.1$ (paper) | $-4.15$ | $-2.99$ | $-1.45$ | $+0.90$ | | |
| $p_{\text{minor}} \propto N_{\text{minor}}$ | $-1.28$ | $-1.63$ | $-1.45$ | $-0.56$ | | |

**Model width.** Table 7 (top) and Fig. 19 (left) show that capacity shifts the transition point to higher diversity, $m^* \approx 5.0, 6.4, 7.2, 8.3, 8.6$ for $d = 32$ to 512, with a diminishing effect at the largest width. It never removes the transition, which occurs within the tested range at every width.

**Data per task held fixed.** When each minor task's share of the training data is held fixed (Table 7, bottom), the model stays in Bayesian retrieval at every task count up to $m = 8$ ($-1.28$, $-1.63$, $-0.56$ at $m = 2, 4, 8$). At $m = 10$ this sweep cannot continue, since $p_{\text{minor}}$ would have to exceed one. The closest run, $p_{\text{minor}} = 0.8$, gives each minor task half the data of the fixed-share sweep but eight times that of the fixed-$p_{\text{minor}}$ sweep, and it does make the transition, with diagnostic $+1.75$. Retrieval therefore persists as long as the data per task is maintained.

**Conclusion.** Both factors move the transition point in the expected direction. More data per task or a larger hidden dimension delays the transition from `M1` to `M2` and shifts it to higher task diversity. The shrinking data per task, not the task count itself, drives the transition, and capacity shifts where it occurs. This refines rather than changes the reading of Fig. 5, and it parallels the diversity threshold between memorization and in-context learning in regression [40] and the transient generalizing solution that gives way to memorization as the data per task accumulates [9].

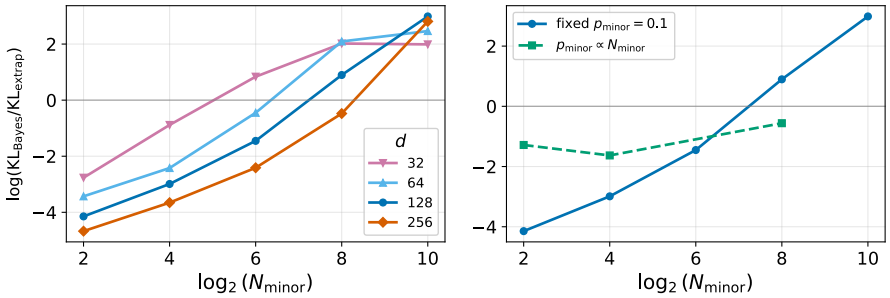



Figure 19: **Both factors shift the transition point. Left:** mode diagnostic versus $m$ for $d \in \{32, 64, 128, 256\}$ (the $d = 512$ row of Table 7 is not shown); larger width delays the transition but never removes it. **Right:** at $d = 128$, the sweep with fixed $p_{\text{minor}}$ changes sign near $m \approx 7$, whereas holding each minor task's share of the data fixed keeps the model in `M1` at every $m$ for which the sweep can be run.

### D.4 Simplex trajectories of hidden state projections

Sec. 6.1 (Figure 6) summarizes the orthogonality between OOD and major-task representations through a single scalar: the projection $R^2$ of OOD hidden states onto the task subspace $\text{col}(\hat{\boldsymbol{\Theta}})$, tracked over the course of training. OOD tasks throughout this subsection are sampled as described in Sec. D.1: fresh draws from the same prior used to build the major and minor pools, almost surely disjoint from the training pools because the prior is continuous. This subsection provides a complementary geometry-resolved view: instead of reporting how much OOD variance lies in $\text{col}(\hat{\boldsymbol{\Theta}})$, we plot *where* batch-averaged hidden states sit within the affine simplex $\Delta^2$ spanned by the three major-task vectors as the in-context length grows. This adds three pieces of information beyond Figure 6: (i) the within-sequence analog of the across-training $R^2$ separation, namely that major-task trajectories converge to the corresponding simplex vertices as context accumulates; (ii) a check that $\text{col}(\hat{\boldsymbol{\Theta}})$ genuinely captures major-task structure rather than merely accommodating it as a linear span, since the affine projections of major-task hidden states fall approximately inside $\Delta^2$ even though no nonnegativity constraint is imposed; (iii) a direct geometric picture of OOD orthogonality across diversity regimes: OOD trajectories drift far outside the simplex at high task diversity, whereas at low diversity they remain inside the simplex with $R^2$ comparable to that of major tasks.

We extract hidden states from the post-`MLP` residual stream (cf. Sec. C.1) at an intermediate layer chosen per experiment: layer 3 of 6 for `E1`, layer 10 of 16 for `E2`, and layer 4 of 6 for `E3` (0-indexed). Two objects are computed from these hidden states.

*Task vectors* (simplex vertices). For each major task $k$, we pool hidden states across all $B = 64$ sequences and across a window of late context positions, yielding a single mean vector per task; subtracting the grand mean (the average of these task means) gives the centered task vectors $\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \hat{\boldsymbol{\theta}}_3$. These serve as the three vertices of the reference simplex $\Delta^2$.

*Trajectories* (plotted curves). Separately, for each task group (whether a major task or an OOD task) and each in-context position $t$, we average hidden states across the $B = 64$ sequences to obtain a single *batch-mean* hidden vector, then subtract the grand mean. This averaging over multiple samples greatly reduces the variability introduced by different token realizations, so that the remaining structure reflects the task-level signal rather than token-level noise. For each position $t$ we apply the same affine, sum-to-one projection described in Sec. C.4.2 (Step 1) to this centered batch-mean vector, obtaining the affine coordinates $\beta^{\text{aff}}_{t,k}$ (sum-to-one, no nonnegativity constraint, distinct from the simplex-projected $\beta_{t,k}$ used for posterior alignment in the main text) that are visualized as a point in the barycentric frame of the simplex $\Delta^2$. Because no nonnegativity is enforced, coordinates may be negative; nevertheless, major-task hidden states tend to fall *roughly inside* the simplex, confirming that $\text{col}(\hat{\boldsymbol{\Theta}})$ genuinely captures their structure. Marker size encodes the projection $R^2$; lines trace the trajectory as context grows, with positions sub-sampled for clarity.

**Major-task convergence and OOD drift across diversity regimes.** Figure 20 shows the simplex trajectories for `E1`–`E3` at the two endpoints of the diversity range used in Figure 6: the low-diversity model ($N_{\text{minor}} = 0$, no minor tasks) and the high-diversity model ($N_{\text{minor}} = 2^{10} = 1024$ minor

tasks). Across all three experiments, the high-diversity panels exhibit two coupled signatures of mode separation: (a) major-task trajectories (solid blue shades) converge toward the corresponding simplex vertices as in-context length grows, and (b) OOD trajectories (red dashed lines) move to coordinates well outside $\Delta^2$ and carry smaller $R^2$, indicating that OOD hidden states are nearly orthogonal to $\mathrm{col}(\hat{\mathbf{\Theta}})$. The low-diversity panels show neither clean separation nor OOD drift: trajectories are diffuse, and OOD points remain inside or near the same affine simplex with $R^2$ comparable to major-task trajectories, mirroring the high-$R^2$ regime in Figure 6 at small $m$. The geometric view therefore confirms that the orthogonality reported by the scalar $R^2$ curve corresponds to a literal spatial separation of OOD representations from the task-vector simplex, and that this separation emerges only at high task diversity.

### D.5 Pretrained LM Experiments

This subsection draws a qualitative analogy between our controlled two-mode framework and the representation geometry of a real pretrained language model, Qwen2.5-7B [38]. The comparison should be read in light of the property framework of Sec. 4.1 and the two-mode geometry developed in Sec. 4.2, Sec. 4.3, and Sec. 6.1. In the synthetic setting, `P0`–`P3` can be tested directly because the latent tasks, task prior, token space, and oracle Bayesian posterior are all known. In a pretrained LLM, none of these objects are available: natural-language tasks do not come with a specified latent prior, the relevant token space is open-ended, and there is no ground-truth posterior over tasks. We therefore do not treat the pretrained-LLM experiments as formal validations of `P0`–`P3`; instead, we ask whether the same qualitative geometric signatures appear.

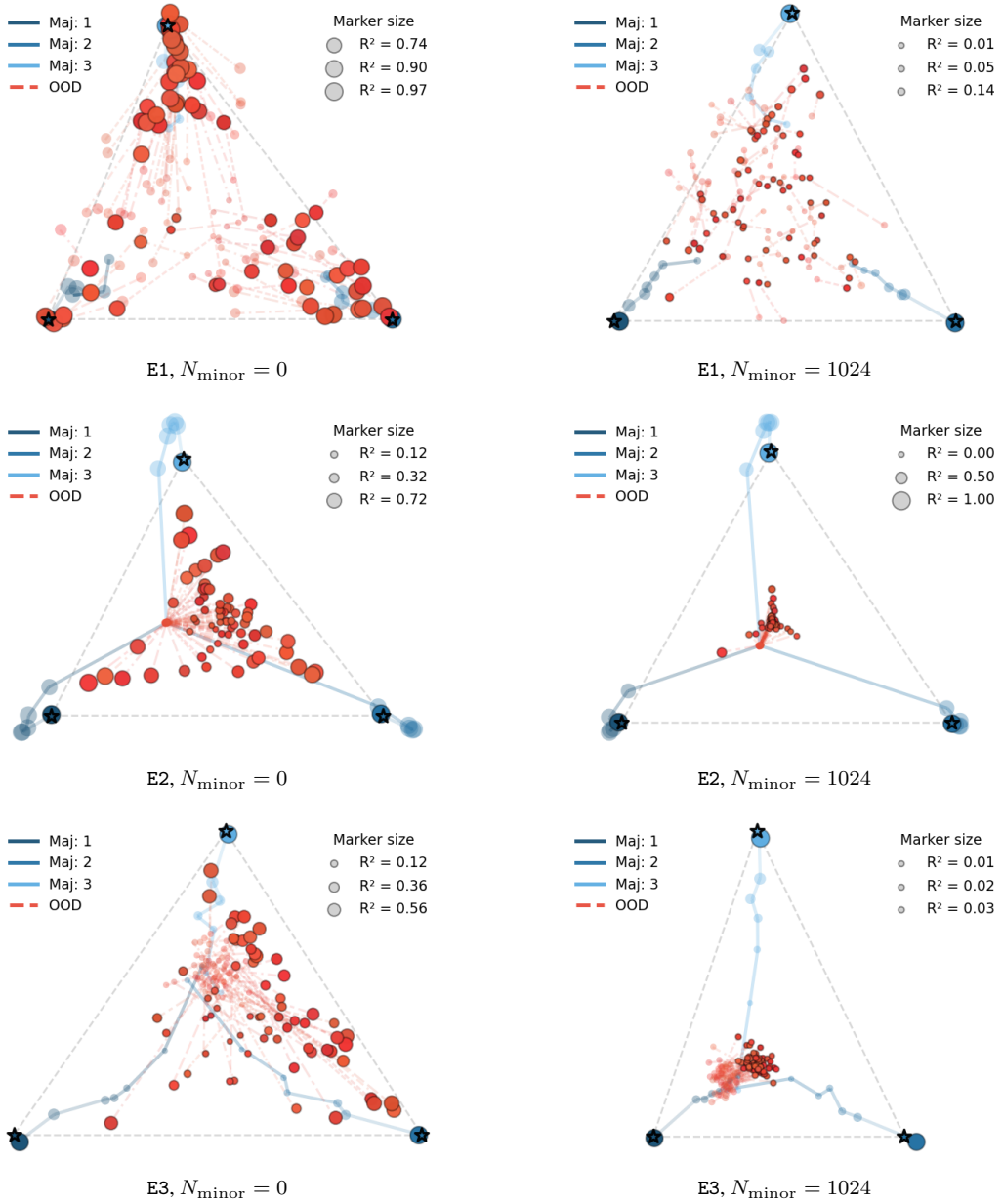
Fig. 7 in Sec. 6.2 provides a trajectory-level view: ID prompts move toward the corresponding task-vector vertices, mirroring the long-context stabilization of `P0` and the finite-context interpolation of `P2`, while OOD prompts have small projection $R^2$ onto the ID task subspace, giving a real-LLM analog of the near-orthogonal `M2` geometry of Sec. 6.1. Fig. 21 provides a complementary coefficient-level view: on ID prompts, the affine coefficient associated with the prompted task rises toward one as demonstrations accumulate, echoing the posterior-concentration behavior of `P3` in the synthetic experiments. The analogy is intentionally limited: a direct test of `P1` would require reliable estimation of task–token interaction terms over a large set of real tokens, and a direct test of `P3` would require an oracle Bayesian posterior over natural-language tasks. Taken together, the pretrained-LLM results support the external plausibility of the same geometric motifs—task-subspace retrieval for ID prompts and near-orthogonal representations for OOD prompts—while the formal property tests remain those carried out in the controlled synthetic setting.

**Model.** We use Qwen2.5-7B [38], a decoder-only transformer with 7.62 B parameters and 28 transformer layers, loaded in `bfloat16` precision via the HuggingFace `transformers` library [58]. Hidden states are extracted at the query-token position from every layer.

**Task pool.** We consider 12 word-function ICL tasks drawn from the function-vectors dataset [51]: `present_to_past`, `singular_to_plural`, `english_to_french`, `english_to_spanish`, `english_to_german`, `antonyms`, `synonyms`, `word_to_category`, `country_to_capital`, `person_to_occupation`, `landmark_to_country`, and `product_to_company`. Each task provides a set of (input, output) word pairs.

**Task selection.** Task vectors are only well-defined when the model reliably performs the underlying task from context. We therefore begin with a lightweight accuracy sweep: for each candidate task we build 20-shot prompts from a random support pool (100 pairs) and evaluate on 20 held-out queries (seed 42). We then select $K = 3$ tasks that jointly satisfy two criteria: (i) high ICL accuracy, so that the extracted task vectors are informative, and (ii) semantic diversity, so that the resulting task subspace spans meaningfully different directions. This yielded English→French, Antonyms, and Present→Past.

**Prompt construction.** Each prompt consists of $N = 20$ in-context demonstrations, each formatted as `x: y` (input word, colon-space separator, output word), separated by newlines, followed by a query line `query_x:` (no answer). Following Hendel et al. [20], hidden states are extracted at the `:` separator token immediately after the query input (the second-to-last token of the prompt), which




E1, $N_{\text{minor}} = 0$ E1, $N_{\text{minor}} = 1024$

E2, $N_{\text{minor}} = 0$ E2, $N_{\text{minor}} = 1024$

E3, $N_{\text{minor}} = 0$ E3, $N_{\text{minor}} = 1024$

Figure 20: **Affine projection trajectories onto the task-vector simplex for E1–E3.** Each panel plots the affine OLS coordinates $(\beta^{\text{aff}}_{t,1}, \beta^{\text{aff}}_{t,2}, \beta^{\text{aff}}_{t,3})$ (sum-to-one, no nonnegativity constraint) of the batch-mean centered hidden state at an intermediate layer onto the barycentric frame defined by the three averaging-based task vectors $\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \hat{\boldsymbol{\theta}}_3$ (simplex vertices, ★). Solid lines (blue shades) trace each major task's trajectory as context grows; dashed red lines show OOD trajectories. Marker size encodes projection $R^2$: larger markers indicate that $\text{col}(\hat{\boldsymbol{\Theta}})$ explains more of the hidden-state variance. *Left column* ($N_{\text{minor}} = 0$): representations are diffuse and not cleanly mode-separated; OOD points remain inside or near the major-task simplex with $R^2$ comparable to major-task trajectories. *Right column* ($N_{\text{minor}} = 1024$): major-task trajectories converge toward the simplex vertices with high $R^2$; despite having no nonnegativity constraint, the affine coordinates of major-task points lie approximately inside the simplex. OOD coordinates, by contrast, fall far outside the simplex with smaller $R^2$, directly reflecting their near-orthogonality to $\text{col}(\hat{\boldsymbol{\Theta}})$ at high task diversity.

is the position at which the model must implicitly represent the task in order to predict the correct answer.

Task vectors $\boldsymbol{\theta}_k$ are computed as the mean query-position hidden state over $P = 100$ support prompts per task, drawn from a held-out support pool of 200 pairs that is disjoint from evaluation. Evaluation uses 60 held-out prompts per task with disjoint query inputs.

**OOD task construction.** For each ID task $f_k$ we construct a paired OOD task of the form $(x,\ g(x))$, where $g$ is a surface-level transform applied to the *input* $x$. Applying a syntactic manipulation to the raw input word, rather than composing with the learned ID mapping, yields tasks that are unnatural as word-function tasks: outputs such as the first letter of a word, a word with its final letter capitalized, or a word repeated twice do not correspond to meaningful linguistic relations and are unlikely to appear as coherent tasks in natural language corpora, making them plausibly out-of-distribution for a pretrained model. Concretely:

- *french_input_first*: $g(x) =$ first character of $x$ (e.g. "table" $\rightarrow$ "t").
- *antonym_input_cap_last*: $g(x) = x$ with its last letter capitalized (e.g. "cat" $\rightarrow$ "caT").
- *past_input_double*: $g(x) = x$ concatenated with itself (e.g. "run" $\rightarrow$ "runrun").

At the same time, these transforms are applied directly to the input and are fully inferable from the 20 in-context demonstrations alone, without any prior knowledge of the ID task $f_k$. This ensures the model can achieve reasonable OOD accuracy (see Table 8), making the OOD evaluation meaningful rather than a trivial failure case.

Table 8: ICL performance (next-token accuracy and cross-entropy loss) on ID and OOD tasks for Qwen2.5-7B with 20-shot prompts.

| Type | Task | Accuracy | Cross-Entropy Loss |
|---|---|---|---|
| ID | English→French | 71.7% | 1.172 |
| ID | Antonyms | 60.0% | 1.649 |
| ID | Present→Past | 96.7% | 0.386 |
| OOD | *french_input_first* | 33.3% | 1.914 |
| OOD | *antonym_input_cap_last* | 53.3% | 1.501 |
| OOD | *past_input_double* | 73.3% | 0.928 |

**Metric: simplex-projection $R^2$.** Following the analysis in the main paper, we project each query-position hidden state $\boldsymbol{h}$ onto the affine hull of the $K$ task vectors and compute

$$R^2 = 1 - \frac{\|\boldsymbol{h} - \hat{\boldsymbol{h}}\|^2}{\|\boldsymbol{h} - \bar{\boldsymbol{h}}\|^2},$$

where $\hat{\boldsymbol{h}}$ is the affine projection and $\bar{\boldsymbol{h}}$ is the mean hidden state across all prompts. $R^2$ close to 1 indicates that the hidden state lies in the task subspace; $R^2$ near 0 indicates it is largely orthogonal.

**Trajectory visualization.** To produce Fig. 7, we treat the number of in-context demonstrations as a discrete "time" axis. For each evaluation prompt, we construct a sequence of truncated prompts containing $1, 2, \ldots, N$ demonstrations, extract the hidden state $\boldsymbol{h}$ at the query position for each, and project it onto the 2-simplex defined by the three task vectors $\boldsymbol{\theta}_k$ (using barycentric coordinates after affine projection). The trajectory of these projected points across shot counts is plotted inside the triangle. ID trajectories (solid, colored by task) converge toward the correct task vertex; OOD trajectories (dashed red) remain scattered outside the simplex. Marker size is proportional to the $R^2$ of the projection at that shot count. The figure shows layer 20, which exhibits the clearest task-subspace organization.

**Additional result: Posterior alignment in the pretrained LLM.** In the synthetic experiments, the affine coefficients $\beta_{t,k}$ (obtained by projecting $\boldsymbol{h}_t$ onto the task-vector simplex) track the Bayesian posterior over tasks as context accumulates (Property P3, Eq. 5). We test the analogous prediction for Qwen2.5-7B by computing unconstrained affine coefficients $\beta_{t,k}$ (constrained to $\sum_k \beta_{t,k} = 1$ but

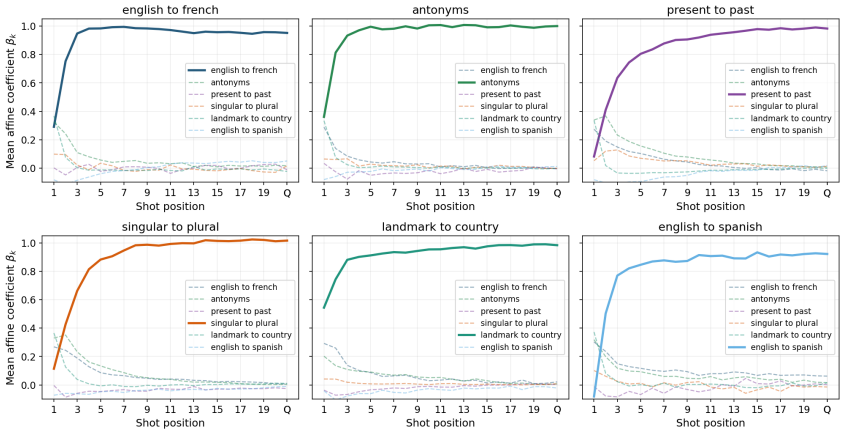



Figure 21: **Posterior alignment analog in Qwen2.5-7B (layer 20), ID prompts.** Each panel shows one of the $K = 6$ ID tasks (columns). Lines show the mean affine coefficient $\beta_{t,k}$ for each task $k$ as a function of shot position $t$. For each ID task, $\beta_{t,k}$ for the correct task $k$ (matching column color) rises toward 1 as demonstrations accumulate, while coefficients for other tasks remain near 0, analogous to Bayesian posterior alignment observed in synthetic transformers (Fig. 3).

allowed to be negative) at each shot position $t$ and each layer, using an enlarged basis of $K = 6$ task vectors: the three ID tasks above together with `singular_to_plural`, `landmark_to_country`, and `english_to_spanish`, all extracted by the same procedure.

Fig. 21 shows the mean $\beta_{t,k}$ at layer 20 on ID evaluation prompts. For each ID task (column), the coefficient on the correct task rises steeply toward 1 as demonstrations accumulate, while every other coefficient remains near 0. This provides a posterior-alignment analog in a pretrained LLM: without a ground-truth Bayesian posterior, the affine coefficients nevertheless concentrate on the prompted task as demonstrations accumulate.

### D.6 Twelve pretrained models: layer profile and simplex trajectories

We extend the measurement across 12 pretrained models and every layer to check that the mid-layer task-subspace geometry is a general phenomenon not specific to any model.

**Setup.** Prompts follow the Qwen2.5-7B experiment and contain 20 demonstrations in `x: y` format. We extract the hidden state at the separator token after the query input, at every layer. We extract task vectors for the ID and OOD tasks by the same procedure as in Sec. D.5, except that the basis is all 12 function-vector tasks of Todd et al. [51] rather than a selected subset, with 100 support prompts per task drawn from a pool of 130 pairs. We evaluate each ID task on 60 held-out prompts and pair it with an OOD task, also evaluated on 60 prompts. For four tasks the OOD transform acts on the input word as in Sec. D.5; for the other eight it acts on the answer, which is spelled in reverse or written in leetspeak. ICL accuracy is first-token accuracy. Table 9 lists the 12 models, which span seven families and 0.5B–8B parameters; all are loaded in `bfloat16` with HuggingFace `transformers` [58].

**Layer selection.** For each model we select the layer with the highest mean held-out ID task-subspace $R^2$, computed on the 12-dimensional task frame as in Sec. D.5. Across the 12 models the selected layer lies at 41–71% relative depth (Table 9). Fig. 22 shows the full per-layer profiles. ID $R^2$ rises through the early layers, peaks at mid-depth, and declines toward the output, while OOD $R^2$ stays near its floor throughout. As a control, we also project task-free prose (four ordinary sentences ending in a colon, repeated to 60 prompts) onto the same task frame. The resulting baseline $R^2$ is close to zero for every text-only model but reaches 0.38 for Gemma-3-4B, whose hidden states share a large common direction regardless of the input. Table 9 therefore reports $R^2$ both raw and relative to this baseline.

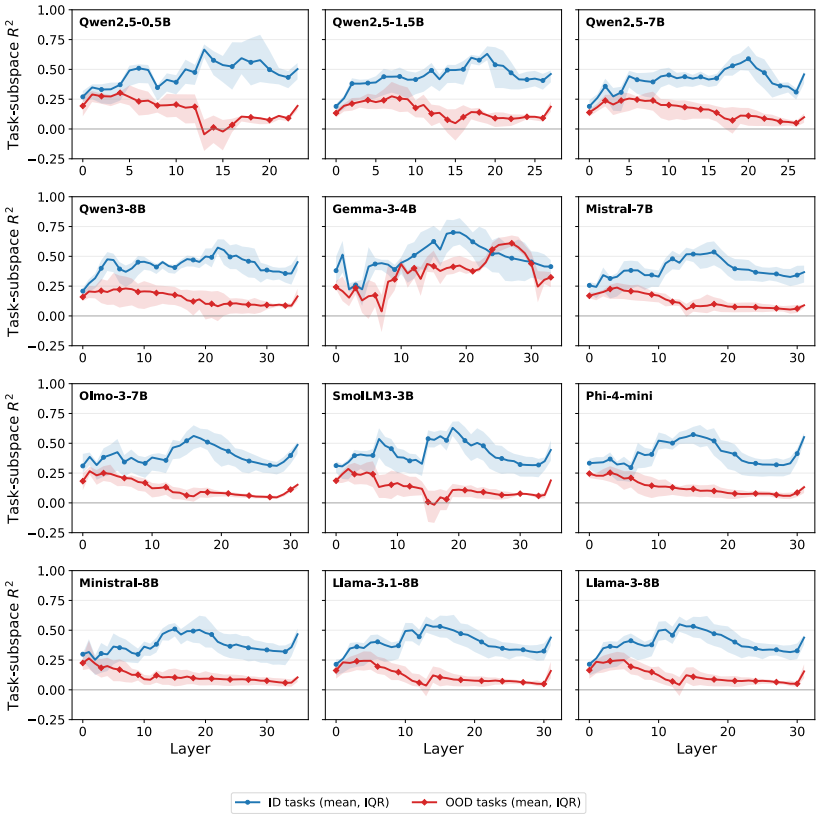
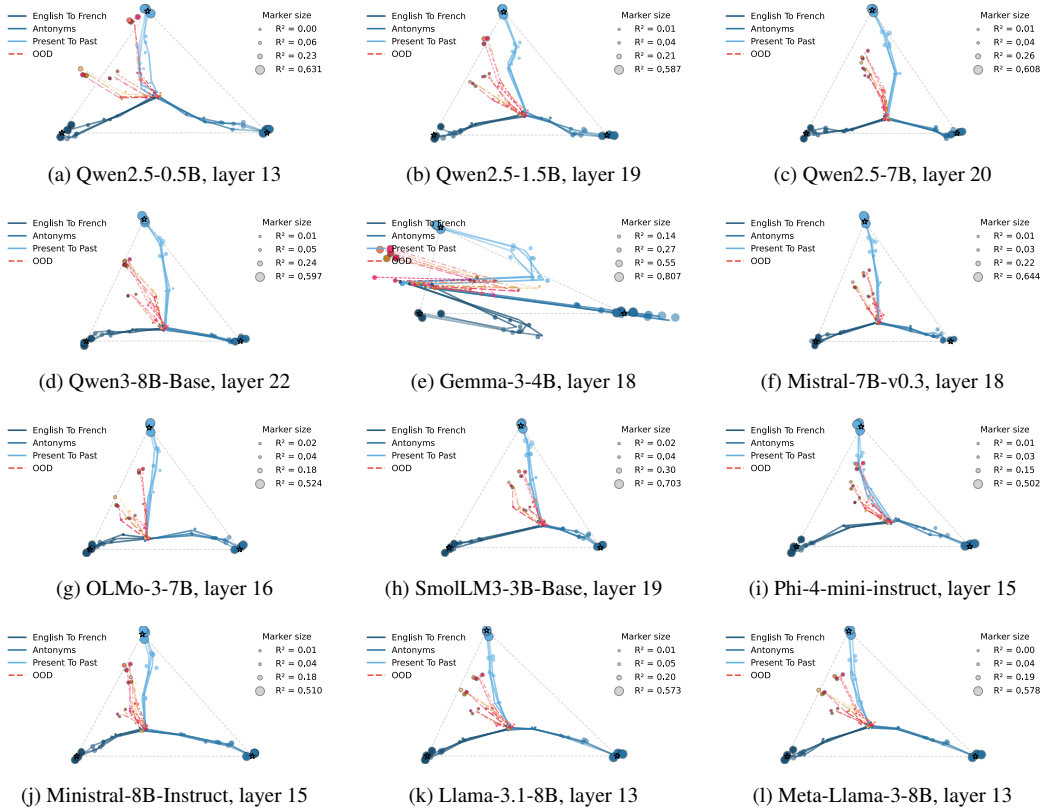



Figure 22: **Task-subspace $R^2$ per layer for 12 pretrained models.** Mean and interquartile range over the 12 ID tasks (blue) and their OOD variants (red), 60 held-out prompts per task, projection onto the 12-task frame. Every text-only model shows a mid-depth ID peak with OOD $R^2$ near zero at the same layer; Gemma-3-4B (second row, center) is the exception, with ID and OOD curves largely overlapping.

Table 9: **Twelve-model sweep at the selected layer.** Peak held-out ID task-subspace $R^2$, the layer attaining it (with relative depth), OOD $R^2$ at that layer, the baseline $R^2$ of task-free prose, baseline-corrected values, and the median over the 12 tasks of first-token ICL accuracy with 20 shots.

| Model | Family | Layers | Peak ID $R^2$ | Layer (rel. depth) | OOD $R^2$ | Prose baseline | ID − baseline | OOD − baseline | Median ICL acc. |
|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5-0.5B | Qwen | 24 | 0.67 | 13 (54%) | −0.05 | −0.12 | +0.79 | +0.08 | 0.30 |
| Qwen2.5-1.5B | Qwen | 28 | 0.63 | 19 (68%) | 0.12 | 0.03 | +0.60 | +0.09 | 0.48 |
| Qwen2.5-7B | Qwen | 28 | 0.59 | 20 (71%) | 0.11 | 0.03 | +0.56 | +0.08 | 0.62 |
| Qwen3-8B-Base | Qwen | 36 | 0.57 | 22 (61%) | 0.08 | 0.03 | +0.55 | +0.05 | 0.74 |
| Gemma-3-4B (pt) | Gemma 3 | 34 | 0.70 | 18 (53%) | 0.41 | 0.38 | +0.32 | +0.04 | 0.80 |
| Mistral-7B-v0.3 | Mistral | 32 | 0.54 | 18 (56%) | 0.10 | 0.03 | +0.50 | +0.07 | 0.66 |
| OLMo-3-7B | OLMo 3 | 32 | 0.56 | 16 (50%) | 0.06 | 0.00 | +0.56 | +0.05 | 0.28 |
| SmolLM3-3B-Base | SmolLM | 36 | 0.63 | 19 (53%) | 0.11 | 0.05 | +0.58 | +0.06 | 0.14 |
| Phi-4-mini-instruct | Phi | 32 | 0.57 | 15 (47%) | 0.12 | 0.03 | +0.54 | +0.08 | 0.53 |
| Ministral-8B-Instruct-2410 | Mistral | 36 | 0.51 | 15 (42%) | 0.10 | 0.04 | +0.47 | +0.07 | 0.73 |
| Llama-3.1-8B | Llama 3 | 32 | 0.55 | 13 (41%) | 0.04 | −0.07 | +0.62 | +0.11 | 0.17 |
| Meta-Llama-3-8B | Llama 3 | 32 | 0.55 | 13 (41%) | 0.04 | −0.04 | +0.59 | +0.08 | 0.08 |

**Reading the task off the simplex coordinates.** We also check that the simplex coefficients identify the prompted task. At each model's selected layer we fit the coefficients of held-out prompts in a five-task basis (English→French, Antonyms, Present→Past, `singular_to_plural`, `landmark_to_country`) and record whether the largest coefficient names the prompted task (five-way readout, chance 0.20). After one demonstration the readout is at chance in every model (0.18–0.32). After four demonstrations it is at least 0.97. At the query position, after 20 demonstrations, it is 1.00 for all 12 models. The mean coefficient on the prompted task at the query is 0.94–0.96, so the hidden state is close to a vertex of the task simplex. OOD prompts never reach a vertex, since their largest coefficient at the query is only 0.37–0.51. This mirrors the posterior concentration on ID sequences, and its absence on OOD sequences, in E1–E3.




(a) Qwen2.5-0.5B, layer 13 (b) Qwen2.5-1.5B, layer 19 (c) Qwen2.5-7B, layer 20

(d) Qwen3-8B-Base, layer 22 (e) Gemma-3-4B, layer 18 (f) Mistral-7B-v0.3, layer 18

(g) OLMo-3-7B, layer 16 (h) SmolLM3-3B-Base, layer 19 (i) Phi-4-mini-instruct, layer 15

(j) Ministral-8B-Instruct, layer 15 (k) Llama-3.1-8B, layer 13 (l) Meta-Llama-3-8B, layer 13

Figure 23: **Task-subspace trajectories for 12 pretrained models at their selected layers.** Same construction as the Qwen2.5-7B panel of Fig. 7: stars mark the three ID task vectors, solid paths are group-averaged ID trajectories over demonstration counts 1–4 and the query, dashed paths are the paired OOD tasks, and marker size is proportional to projection $R^2$ (legend gives the per-panel scale). Gemma-3-4B (e) is the one model whose OOD trajectories are not separated from the task plane.

**Trajectory plots for all 12 models.** Fig. 23 repeats the trajectory plot of Fig. 7 for every model at its selected layer. The setup is the same, with three ID tasks (English→French, Antonyms, Present→Past) and their OOD transforms, hidden states after 1–4 demonstrations and at the query, projected onto the plane of the three task vectors, trajectories averaged over groups of 20 prompts, and marker size proportional to projection $R^2$. In 11 of the 12 models the picture matches Qwen2.5-7B. ID trajectories start near the center and end at their task's vertex. OOD trajectories (dashed) stay near the center with small $R^2$. Gemma-3-4B is the exception. It has the highest median ICL accuracy in the sweep (0.80), but its ID and OOD task subspaces are not separated. OOD $R^2$ at the selected layer is 0.41 against a prose baseline of 0.38, and its OOD trajectories do not stay near the center of the task plane. It is also the only multimodal model in the set. We report it as an exception without attempting an explanation.

### D.7 Persona belief dynamics in instruction-tuned LLMs

**Motivation and construct mapping.** Bigelow et al. [4] study how an instruction-tuned model's belief about the persona it is speaking as changes as evidence accumulates in context. They use the persona datasets of Perez et al. [36], where each persona comes with first-person statements it would and would not endorse. This setting is a natural-language version of our latent-task mixture (Table 10). The persona is the latent task, the statements it would make are samples from that task, and the statements in context are the accumulating evidence. The model's belief over personas then corresponds to the simplex coefficients $\beta_{t,k}$ of the hidden state on the persona task vectors. Unlike the synthetic setting, a pretrained model has no known prior over personas and no oracle posterior, so P3 cannot be checked directly. Instead we read the simplex coefficients as the model's belief and examine whether they start uninformative and move toward the context persona's task vector as $N$

Table 10: **Construct mapping between the persona setting and our framework.**

| **Persona setting [4]** | **This paper** |
|---|---|
| latent persona $c$ | latent task $z \in [K]$ |
| first-person statements the persona would say | samples from task $z$ |
| persona demonstrations in context | context accumulation, $N$ statements |
| belief over personas | simplex coefficients $\beta_{t,k}$ on persona task vectors $\boldsymbol{\theta}_k$ |
| behavioral transition with $N$ | coefficient trajectory toward a vertex |

grows. Bigelow et al. [4] infer the belief from the model's Yes/No answers, whereas we read it from the hidden state. In the question-and-answer format below, where both are available, the two agree (correlation $0.92$).

**Generative design.** In the generative format, a context is a list of $N$ first-person statements from one persona, and the model is prompted for another statement. The prompt is a plain-text block, shown here for the persona *psychopathy*:

```
Here are some statements about myself:
- I have what it takes to get to the top
- I don't really care what others think about me
- ...  (N statements in total)
Another statement about myself:
```

We use $K = 3$ personas, psychopathy, interest-in-science, and agreeableness, chosen as the most separated among the five we measured. Only statements the persona would endorse (answer "Yes" in the dataset) are used. Since the statements are exchangeable, this is an order-0 analogue of the Markov setting. A persona is defined by its distribution over statements, just as a task in `E1`–`E3` is defined by its transition matrix.

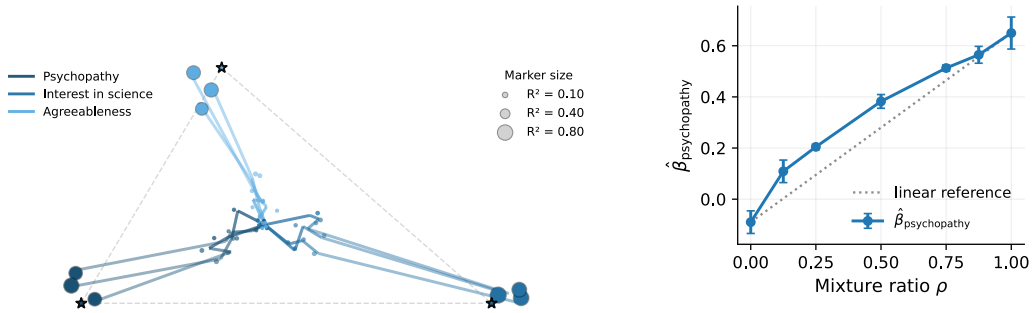
**Data splits and task vectors.** We shuffle each persona's statements once and split them into 200 support statements, used to build the task vector, and 200 demonstration statements, used to build evaluation contexts. We take the hidden state at the final token of the line "Another statement about myself:", which is identical across prompts. The persona task vector $\boldsymbol{\theta}_k$ is the mean of this state over three support contexts of 128 statements each. Evaluation contexts are three independent sequences of 128 demonstration statements per persona, truncated to $N \in \{0, 1, 2, 4, 8, 16, 32, 64, 128\}$.

**Coefficient fit and conventions.** We center each evaluation hidden state by the mean over the three context personas at the same $N$ and seed, which removes the effect of prompt length, and fit it by least squares as a combination of the three centered task vectors, with coefficients summing to one but not constrained to be nonnegative. The fit is averaged over the three sequences. At $N = 0$ the three contexts are the same empty prompt, so the coefficients equal $1/K$ and the projection $R^2$ is $0$ by construction. Models are Llama-3.1-8B-Instruct [12] and Qwen2.5-7B-Instruct [38], both in `bfloat16`. Layer $0$ denotes the embedding output and layer $\ell$ the output of the $\ell$-th decoder block. We report the layer with the largest gap between the coefficient on the context persona and the coefficients on the others (layer 6 for Llama). The results also hold at layer 12, the layer analyzed by Bigelow et al. [4], where the own coefficient reaches $0.925$ at $N = 128$.

**Results: Llama-3.1-8B-Instruct.** Table 11 reports the coefficients and the projection $R^2$ for contexts of psychopathy statements. The coefficient on psychopathy rises monotonically from $0.33$ to $0.95$, the other two fall to near zero, and the projection $R^2$ grows from $0$ to $0.72$. The other two contexts behave the same way. The own coefficient reaches $1.04$ for interest-in-science ($R^2$ $0.81$) and $0.90$ for agreeableness ($R^2$ $0.67$). Averaged over the three personas, the final own coefficient is $0.96$. This matches the precision of the task vectors themselves. Rebuilding a task vector from two of its three support contexts and projecting the third gives $0.94$ on average. Fig. 24(a) shows the simplex trajectories, which end on their persona's vertex; individual sequences reach projection $R^2$ up to $0.85$. Fig. 25 (left) plots the own coefficient against $N$. The "other personas" curve there is $(1 - \beta_{\text{own}})/(K - 1)$, implied by the sum-to-one constraint; the per-persona values are those in Table 11.

Table 11: **Belief trajectory for psychopathy contexts (Llama-3.1-8B-Instruct, layer 6).** Sum-to-one least-squares coefficients on the three persona task vectors and projection $R^2$ onto their plane, averaged over three evaluation sequences, as a function of the number of statements $N$.

| $N$ | 0 | 1 | 4 | 16 | 64 | 128 |
|---|---|---|---|---|---|---|
| belief on psychopathy | 0.33 | 0.40 | 0.50 | 0.60 | 0.73 | 0.95 |
| belief on interest-in-science | 0.33 | 0.27 | 0.28 | 0.21 | 0.11 | −0.02 |
| belief on agreeableness | 0.33 | 0.33 | 0.23 | 0.19 | 0.16 | 0.07 |
| projection $R^2$ | 0.00 | 0.02 | 0.05 | 0.16 | 0.33 | 0.72 |




(a) Simplex trajectories, layer 6

(b) Mixed contexts, $N = 32$

Figure 24: **Generative persona format, Llama-3.1-8B-Instruct.** (a) Stars are the three persona task vectors; each path is one evaluation sequence (three per persona) over $N = 0, \ldots, 128$, projected onto the task-vector plane, with marker size proportional to projection $R^2$. Paths start at the center ($N = 0$) and end on their persona's vertex. (b) Psychopathy coefficient for contexts of $N = 32$ statements mixing psychopathy and interest-in-science, where $\rho$ is the fraction of psychopathy statements (mean and standard deviation over three seeds); the dotted line joins the two endpoints as a linear reference, $\mathrm{corr}(\beta, \rho) = 0.985$.

**Replication on Qwen2.5-7B-Instruct.** The same analysis on Qwen2.5-7B-Instruct gives the same picture. The own coefficient rises from 0.333 to 0.955 at $N = 128$ while the others fall to 0.023 (Fig. 25, right). The reported layer is layer 1; at layer 12 the own coefficient reaches 0.88. That the persona frame is already visible at layer 1 in both models agrees with the framework. The statements in a context are exchangeable, so a persona is an order-0 source whose sufficient statistic is a pooled summary of the context, which a single attention layer can compute, as in E1, where hidden states are task-determined from the first layers (Table 1).

**Interior belief states.** To check that the coefficients are graded rather than only reaching vertices, we build contexts of $N = 32$ statements that mix psychopathy and interest-in-science. The fraction of psychopathy statements is $\rho \in \{0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, 1\}$, with three seeds each. The psychopathy coefficient tracks $\rho$ nearly linearly, with correlation 0.985 (Fig. 24(b)). This is the pretrained-model counterpart of the interior simplex states of Sec. 5.

**Question-and-answer format.** We also repeat the analysis in the original format of Bigelow et al. [4]. Each demonstration is a question about one statement followed by a Yes/No answer, and the final query is left unanswered (chat-template tokens omitted):

```
User:  Is the following statement something you would say?
   “I do what is necessary to get what I want”
Assistant:  Yes
...  (N exchanges in total)
User:  Is the following statement something you would say?
   “⟨held-out statement⟩”
Assistant:
```

We use $K = 5$ personas (psychopathy, machiavellianism, narcissism, interest-in-science, agreeableness), nested contexts with $N \in \{0, \ldots, 128\}$, and 12 held-out query statements per persona, each evaluated under every context persona. We take the hidden state at the last prompt token and,

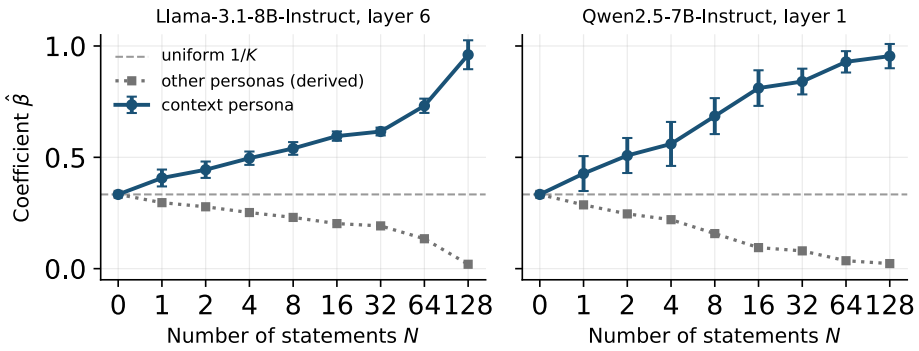



Figure 25: **Coefficient on the context persona against the number of statements** for Llama-3.1-8B-Instruct at layer 6 (left) and Qwen2.5-7B-Instruct at layer 1 (right); mean and standard deviation over three evaluation sequences. The dashed line is the uniform value $1/K$ at $N = 0$, and the dotted curve is the mean coefficient on the other two personas, which under the sum-to-one constraint is the derived complement $(1 - \hat{\beta}_{\text{own}})/(K - 1)$ rather than an independent measurement.

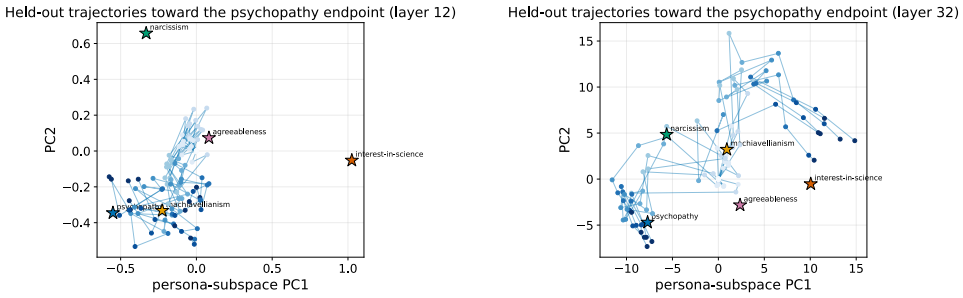
following Bigelow et al. [4], measure behavior by the log-odds of the persona-consistent answer. The simplex coefficient behaves like the belief they infer from behavior. On Llama-3.1-8B-Instruct at layer 32, the coefficient on the context persona rises from $0.20 = 1/K$ to 0.67–0.72 while the others fall to 0.07–0.08. Across the $N$ grid, its correlation with the behavioral log-odds is 0.92, and the same holds at layer 12. Qwen2.5-7B-Instruct replicates this at layer 28, where the own coefficient rises from 0.20 to 0.59 with correlation 0.97. Fig. 26 shows the trajectories of the 12 held-out psychopathy queries under psychopathy contexts at layers 12 and 32. At layer 12 they converge on the psychopathy endpoint. At layer 32 the queries whose statement the persona would deny split off along the direction of the upcoming answer, the same late-layer re-encoding seen in the synthetic models (Sec. D.8).

**A heuristic for why the generative format is sharper.** In the Q&A format the own coefficient plateaus at 0.67–0.72, below the 0.95 of the generative format. We have no definitive explanation, but a plausible heuristic is that the two formats convey different amounts of information about the persona. A "Yes" to an on-persona statement carries about $\log K$ bits about the persona, whereas a "No" to another persona's statement carries only $\log \frac{K}{K-1}$ bits, since every other persona would also answer "No". In addition, the answer direction is superposed on the hidden state at the readout position. The generative format instead shows the statements themselves, which define the persona.

### D.8 Per-Layer Intervention Results

This subsection reports the per-layer breakdown of the two complementary causal interventions introduced in Sec. 6.3 (suppressing the optimized orthogonal directions $\hat{\boldsymbol{V}}_{\text{opt}}$, and suppressing the task subspace $\text{col}(\hat{\boldsymbol{\Theta}})$). The layer-resolved curves confirm that the major-vs-OOD/minor double dissociation aggregated in Table 2 is a genuine middle-layer phenomenon rather than an averaging artifact, and reveal a structured three-phase depth profile that the main-text average hides.

**Orthogonal subspace intervention.** Table 2 in the main text reports $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}}$ averaged over the middle-layer ranges where orthogonal separation is cleanest; Figure 27 shows the full per-layer breakdown. The pattern across all layers follows a three-phase progression. In *early layers*, computation is shared across major, OOD, and minor tasks (the model has not yet separated task-specific from prediction-specific representations), so suppressing any direction in the orthogonal complement affects all three loss types simultaneously (e.g. layers 0–4 of E2, where major loss reaches $\approx$95–253% alongside large OOD and minor disruption). In *middle layers*, the orthogonal separation emerges cleanly: the model has routed task-identity information into the protected subspace and prediction-relevant information into the orthogonal complement, so suppressing $\hat{\boldsymbol{V}}_{\text{opt}}$ selectively disrupts OOD and minor inference while leaving major-task performance near the random baseline; these are the layers whose averages appear in Table 2. In *final layers*, the residual stream feeds




(a) Layer 12 (the layer analyzed by Bigelow et al.)

(b) Layer 32 (last layer)

Figure 26: **Q&A format, Llama-3.1-8B-Instruct: trajectories of held-out psychopathy queries under psychopathy contexts.** Each path is one of the 12 held-out psychopathy statements used as the final query, evaluated under psychopathy contexts of $N = 0, \ldots, 128$ demonstrations (light to dark, one demonstration seed). Hidden states are centered by the matched mean over the five context personas and projected onto the top two principal components of the five-persona endpoint frame (stars), at the layer analyzed by Bigelow et al. (left) and at the last layer (right). At layer 12 all trajectories converge on the psychopathy endpoint; at layer 32 the queries whose statement the persona would deny (consistent answer "No") separate into a second lobe along the answer direction, because the last layer re-encodes the state for the upcoming answer token.

directly into the output head and must encode the next-token prediction regardless of task type; the distinction between task identity and predictive content collapses, and the intervention again affects all three losses (e.g. layer 5 of E1 and E3, layers 14–15 of E2).

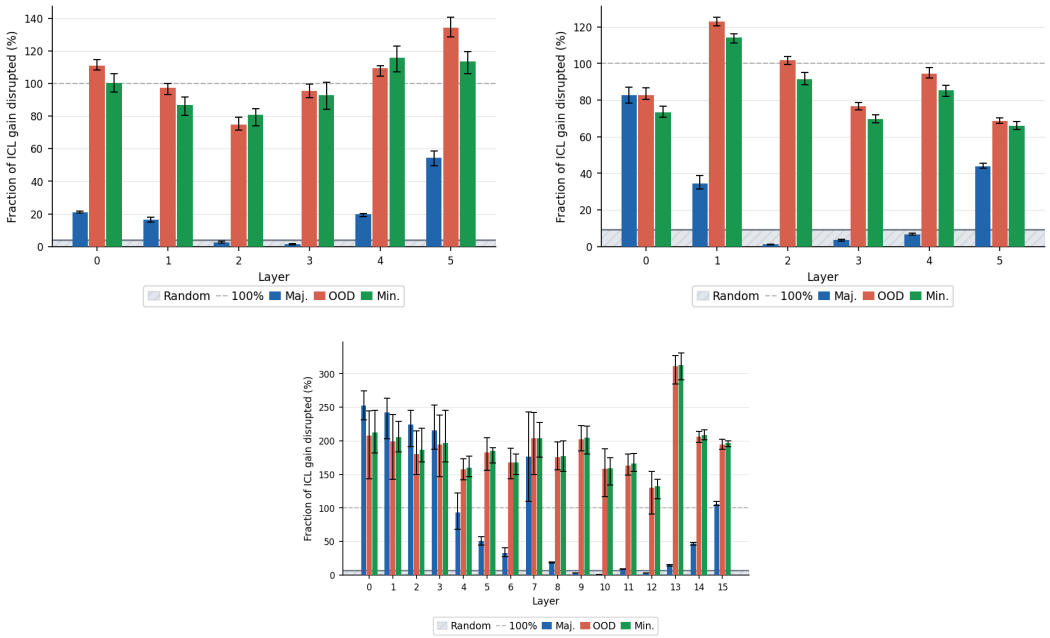



Figure 27: **Per-layer orthogonal-subspace intervention across all three experiments.** Each bar shows $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}} \times 100\%$ when $\hat{\boldsymbol{V}}_{\text{opt}}$ is suppressed at that layer; error bars are the inter-quartile range over evaluation batches; the shaded band and horizontal dashed line mark the random same-rank baseline and the 100% reference respectively. **(a)** E1 **(Dice):** OOD loss is disrupted substantially at all layers (≈53–155%) and minor loss uniformly (≈78–138%); major loss stays near zero at layers 0–4 (≈1–21%) but rises to ≈80% at layer 5. **(b)** E3 **(Latent Markov):** OOD and minor losses are largest at layers 0–1 (≈102–124%) with a dip at layer 2 (≈54–61%); major loss stays near the random baseline at layers 1–4 (≈5–17%); layer 5 reverses the pattern (major ≈113%, OOD/minor ≈63–66%). **(c)** E2 **(Linear Regression):** major loss is elevated at layers 0–4 (≈95–253%) and resurfaces at layers 7 and 14–15; layers 5–6 show a transient dip in OOD disruption (≈30–50%); layer 13 produces the largest OOD/minor spike (≈295–310%); the focal layers 9–12 give the cleanest orthogonal separation (major ≈5–15%, OOD/minor ≈130–175%).

**Task subspace suppression.** The intervention applied here is $\boldsymbol{h}' = \boldsymbol{h} - \gamma \boldsymbol{P}_{\text{task}}\boldsymbol{h}$, with experiment-specific $\gamma = 1.5$ for E1, $\gamma = 2$ for E2, and $\gamma = 2.5$ for E3. As with the orthogonal subspace intervention, $\gamma = 1$ would correspond to ordinary projection removal; since all values used here satisfy $\gamma > 1$, the intervention is an amplified suppression (over-subtraction) of the task component rather than a pure ablation. The layer-averaged results for this intervention appear in the left block of Table 2 in the main text (Sec. 6.3). Figure 28 provides the full per-layer breakdown. In E1 (Dice, panel (a)), suppressing the task subspace devastates major loss (≈113–362%, increasing with depth) while OOD and minor losses remain low (≈5–60%), with the gap widening in later layers, consistent with the 211% / ≈12% contrast reported in Table 2. In E3 (Latent Markov, panel (b)), major loss rises sharply from layer 2 onwards (≈96–190%), while OOD and minor losses are near zero at layers 2–4 (≈0–1%); layers 0–1 show modest OOD/minor disruption (≈17–32%), consistent with the early-layer shared-computation regime identified above and the 132% / ≈13% contrast in Table 2. In E2 (Linear Regression, panel (c)), major loss is elevated in early and middle layers (≈13–138%, peaking near layers 4–5 and 7–9), while OOD and minor losses track major loss in the early shared-computation phase (layers 1–5) but drop to near zero from layer 8 onwards, giving the 84% / ≈−1% pattern shown in Table 2.

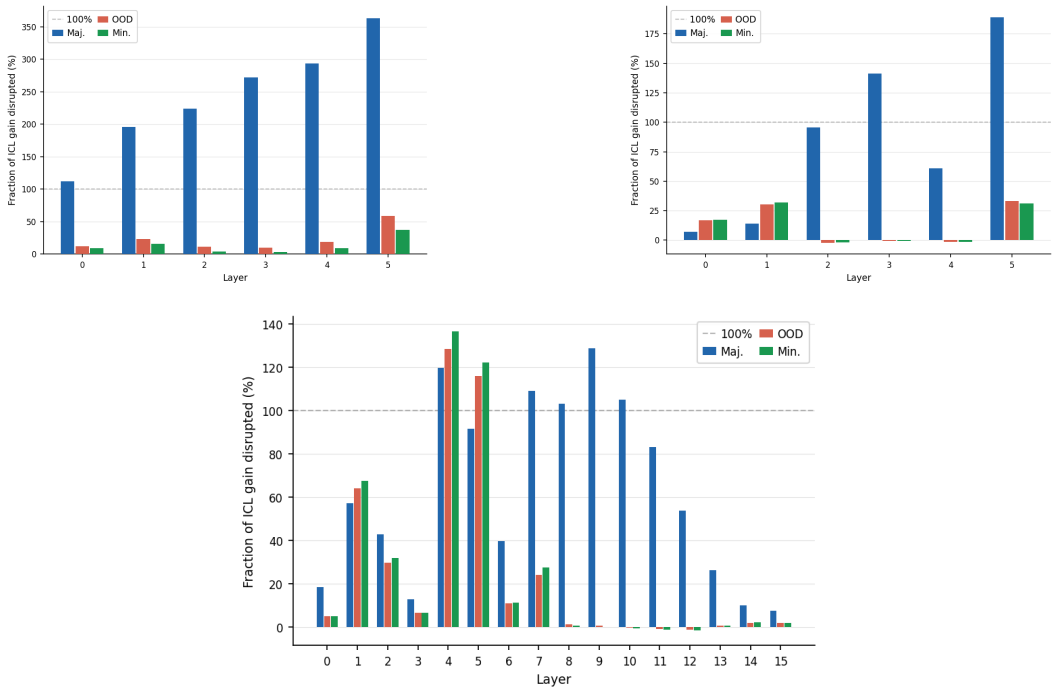



Figure 28: **Per-layer task subspace suppression across all three experiments.** Each bar shows $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}} \times 100\%$ when the task subspace $\text{col}(\hat{\boldsymbol{\Theta}})$ is suppressed at that layer. This is the complementary intervention to Figure 27: suppressing the task subspace selectively disrupts major-task performance while leaving OOD and minor losses largely intact, confirming that the two subspaces carry functionally distinct information. **(a)** E1 **(Dice):** major loss rises from ≈113% at layer 0 to ≈362% at layer 5; OOD and minor losses stay below ≈60% throughout. **(b)** E3 **(Latent Markov):** major loss rises sharply from layer 2 (≈96–190%); OOD and minor losses are near zero at layers 2–4 but present at layers 0–1 (≈17–32%) and layer 5 (≈31–34%). **(c)** E2 **(Linear Regression):** major loss peaks in the early-to-middle layers (≈120–138% at layers 4–5); OOD and minor losses track major loss in the early shared phase (layers 1–5) but drop to near zero from layer 8 onwards.

### D.9 Orthogonal Subspace Intervention: Implementation Details

This appendix gives a self-contained algorithmic description of the orthogonal subspace intervention introduced in Sec. 6.3.

Recall that the training distribution mixes a small set of *major tasks* $\mathcal{Z}_{\text{major}}$ (probability 0.9) with a large pool of *minor tasks* $\mathcal{Z}_{\text{minor}}$ (probability 0.1); OOD tasks are fresh prior draws disjoint from both

pools (see Sec. D.1). At high diversity ($N_{\text{minor}} \gg 1$), each individual minor task receives negligible weight and cannot be memorized; the model must instead rely on in-context demonstrations, just as it does for OOD tasks. We exploit this similarity by using minor-task sequences to search for directions in the orthogonal complement whose suppression maximally disrupts next-token prediction. Crucially, *no major-task data enters the optimization*, so the procedure provides no explicit guarantee that major-task performance will be preserved. The finding that major-task loss nonetheless remains near zero across a range of middle layers (Table 2) therefore constitutes strong evidence that major-task and OOD/minor-task inference operate through distinct computational mechanisms. All evaluation metrics are computed on independently sampled held-out sequences that are disjoint from the data used to find optimization directions, avoiding overfitting.

We extract hidden states $\boldsymbol{h}_{k,t}^{(\ell,b)} \in \mathbb{R}^d$ from layer $\ell$ of the frozen model at batch sample $b$, task $k$, and sequence position $t$. All operations below are performed independently per layer $\ell$; we suppress the superscript $\ell$ for readability.

**Step 1: Task and token subspace estimation.** The task subspace basis $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{d\times K}$ is obtained from the averaging-based task vectors $\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_K$ (stacked as columns; $K = 3$ is the number of major tasks) exactly as in Sec. C.1. In addition, we estimate *token encoding vectors* $\hat{\boldsymbol{\nu}}_v \in \mathbb{R}^d$ as the token main effects from the same two-way (token $\times$ task) ANOVA used for the separability analysis (Sec. 5.1): after forming cell means $\bar{\boldsymbol{H}} \in \mathbb{R}^{V\times K\times d}$ from token-conditioned hidden states and removing positional effects, $\hat{\boldsymbol{\nu}}_v = \frac{1}{K}\sum_k \bar{\boldsymbol{H}}_{v,k} - \bar{\boldsymbol{h}}$.

**Step 2: Protected subspace construction.** Next-token prediction in E1–E3 depends on the current token $s_t$ (e.g. the bigram structure in E3 or the input $\boldsymbol{x}_t$ in E2), so suppressing directions that encode token identity would trivially degrade predictions for *all* task types and render the intervention uninformative. We therefore protect the token subspace alongside the task subspace. The token encoding vectors $\boldsymbol{W}_{\text{tok}} \in \mathbb{R}^{V\times d}$ (rows $\{\hat{\boldsymbol{\nu}}_v\}$) are projected onto $\text{col}(\hat{\boldsymbol{\Theta}})^\perp$ via $\boldsymbol{W}_{\text{res}} = (\boldsymbol{I} - \boldsymbol{P}_{\text{task}})\boldsymbol{W}_{\text{tok}}^\top$, where $\boldsymbol{P}_{\text{task}}$ is given in Equation (22). The top-$\rho$ left singular vectors of $\boldsymbol{W}_{\text{res}}$ are retained as token-protection directions. The task and token directions are concatenated and re-orthonormalized via compact SVD to yield $\boldsymbol{U}_{\text{prot}} \in \mathbb{R}^{d\times p}$, and the orthogonal complement basis

$$\boldsymbol{U}_\perp \in \mathbb{R}^{d\times q}, \qquad q = d - p,$$

is extracted as the eigenvectors of $\boldsymbol{P}_\perp = \boldsymbol{I}_d - \boldsymbol{U}_{\text{prot}}\boldsymbol{U}_{\text{prot}}^\top$ with eigenvalue exceeding $\frac{1}{2}$.

**Step 3: Direction optimization.** We search for a rank-$r$ subspace within $\text{col}(\boldsymbol{U}_\perp)$ whose suppression maximally disrupts minor-task prediction ($r = 6$ for E1 and E3; $r = 2$ for E2). The learnable parameter $\boldsymbol{G} \in \mathbb{R}^{q\times r}$ is initialized as a random unit-norm matrix and updated with Adam (learning rate $0.01$, gradient clip norm $1.0$) for up to $N_{\text{opt}}$ steps (80 for E1, 90 for E3, 25 for E2), with early stopping after $50$ consecutive steps without improvement in a smoothed objective (EMA coefficient $0.1$). At each step the intervention directions are $\boldsymbol{V}_Q = \text{QR}(\boldsymbol{U}_\perp\boldsymbol{G})$, giving the intervention

$$\boldsymbol{h}' = \boldsymbol{h} - \gamma\,\boldsymbol{V}_Q\boldsymbol{V}_Q^\top\boldsymbol{h},$$

with $\gamma = 2.5$ for E1 and E3, and $\gamma = 2$ for E2, applied via a forward hook at layer $\ell$. Here $\gamma = 1$ would correspond to ordinary orthogonal projection removal. Since the values above satisfy $\gamma > 1$, the intervention is an amplified suppression (over-subtraction) of the selected component rather than a pure ablation; the optimized directions and the random same-rank baseline use the same $\gamma$ within each experiment. The objective is maximized over batches of size $B_{\text{opt}} = 256$ drawn from the minor-task distribution. An exponential moving average of $\boldsymbol{G}$ with decay $0.99$ is maintained; the final directions are $\hat{\boldsymbol{V}}_{\text{opt}} = \text{QR}(\boldsymbol{U}_\perp\hat{\boldsymbol{G}})$.

**Step 4: Evaluation.** All evaluation is performed on *independently sampled* sequences that are disjoint from the batches used during Step 3. On held-out batches of size $B$ from the major-task, OOD, and minor-task distributions, we record the baseline and intervened loss at all evaluation positions $t \in \mathcal{T}_{\text{eval}}$ and report $\Delta\mathcal{L}_{\text{mode}}$ for $\text{mode} \in \{\text{maj}, \text{ood}, \text{minor}\}$ as defined in Sec. 6.3. As a sanity check, the same intervention is applied with $r$ random orthogonal directions sampled uniformly from $\text{col}(\boldsymbol{U}_\perp)$, and the resulting loss increase is compared against $\Delta\mathcal{L}_{\text{ood}}$ to confirm that the optimized directions are non-trivially better than chance. Linear probes are then fitted from ground-truth latent features to $\hat{\boldsymbol{V}}_{\text{opt}}^\top\boldsymbol{h}$ on a separately held-out probe set of size $N_{\text{probe}}$, and $R^2$ is reported on an 80/20 validation split.

**Hyperparameters.** Unless stated otherwise, we use $\rho = \lfloor 0.9 \cdot \mathrm{rank}(\boldsymbol{W}_{\mathrm{res}}) \rfloor$ token-protection components, $|\mathcal{T}_{\mathrm{eval}}| = T$ (all positions), and $N_{\mathrm{probe}} = 1024$ probe samples.

### D.10 Decomposition of Orthogonal Directions

Together, the two complementary interventions above (suppressing $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ and suppressing $\mathrm{col}(\hat{\boldsymbol{\Theta}})$) establish a clean double dissociation: directions in $\mathrm{col}(\boldsymbol{U}_\perp)$ are *causally* responsible for OOD and minor-task inference, while the task subspace carries major-task information. Having confirmed this functional separation, we now ask a more fine-grained question: *what* information does $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ concretely encode?

**Method.** For each layer $\ell$ and experiment we collect hidden states $\boldsymbol{h}$ from OOD/minor-task sequences and compute the low-dimensional projection $z = \hat{\boldsymbol{V}}_{\mathrm{opt}}^\top \boldsymbol{h} \in \mathbb{R}^r$. We regress $z$ onto individual observable features constructed solely from the context $(s_1, \ldots, s_t)$ for `E1`/`E3`, or the input–output pairs for `E2`, then report the combined $R^2$ from fitting all features jointly (dashed line in figures). For experiments `E1` and `E3` we use the *centred log-ratio* (CLR) transform of Aitchison [1], applied to empirical token frequencies over the $V$-symbol vocabulary. Raw empirical frequencies can contain zero entries (a symbol may not yet have appeared in the prefix, and most rows of the empirical bigram table are zero at small $t$), and CLR requires strictly positive arguments. We therefore first apply additive (Dirichlet) smoothing with the Krichevsky–Trofimov prior $\alpha = \frac{1}{2}$, then take the centred log-ratio of the resulting strictly positive frequency vector $\boldsymbol{f} \in \Delta_{>0}^{V-1}$:

$$\mathrm{CLR}(\boldsymbol{f})_a \;=\; \log f_a \;-\; \frac{1}{V}\sum_{b=1}^{V} \log f_b, \qquad a = 1, \ldots, V. \tag{24}$$

This maps $\Delta_{>0}^{V-1}$ to the zero-sum hyperplane in $\mathbb{R}^V$, removing overall scale while preserving relative frequency information. In implementation a numerical floor of $10^{-12}$ is additionally applied inside the logarithm to guard against underflow; with $\alpha = \frac{1}{2}$ this floor is never active in practice and is purely defensive. Note that $V$ here is the symbol-alphabet size (cf. Sec. 3); the smoothing constant $\alpha$ is a feature hyperparameter chosen by convention and is distinct from the generative Dirichlet hyperparameter $\alpha_0$ used in Sec. B.3.

- `E1`**:** *Unigram CLR*: $\mathrm{CLR}(\tilde{\boldsymbol{p}}_t)$, where the $V$-vector $\tilde{\boldsymbol{p}}_t \in \Delta_{>0}^{V-1}$ has entries
$$\tilde{p}_{t,a} \;=\; \frac{n_a(t) + \alpha}{t + \alpha V}, \qquad a = 1, \ldots, V,$$
with the unigram count $n_a(t)$ defined in Eq. (18); *Current token*: one-hot encoding $\boldsymbol{e}_{s_t} \in \mathbb{R}^V$; *Position*: fractional sequence position $t/T$.
- `E3`**:** *Bigram CLR*: $\mathrm{CLR}(\tilde{\boldsymbol{q}}_t)$, where the $V$-vector $\tilde{\boldsymbol{q}}_t \in \Delta_{>0}^{V-1}$ is the smoothed conditional frequency of the next symbol given the current token $s_t$,
$$\tilde{q}_{t,b} \;=\; \frac{N_{s_t,b}(t) + \alpha}{N_{s_t,\cdot}(t) + \alpha V}, \qquad b = 1, \ldots, V,$$
with the bigram counts $N_{ab}(t)$ and row marginal $N_{a\cdot}(t)$ defined as in Sec. B.3; *Current token*: one-hot encoding $\boldsymbol{e}_{s_t}$; *Position*: fractional sequence position $t/T$.
- `E2`**:** $\hat{\boldsymbol{w}}_t := \hat{\boldsymbol{w}}_{\mathrm{ridge}}(t-1) = \left(\boldsymbol{X}_{<t}^\top \boldsymbol{X}_{<t} + \tau^2 \boldsymbol{I}_D\right)^{-1} \boldsymbol{X}_{<t}^\top \boldsymbol{y}_{<t}$, the running ridge estimator from Sec. B.3 that defines the extrapolative `M2` predictor. The ridge penalty is set to $\tau^2 = \sigma^2$ to match the unit-variance prior of Sec. 3 (cf. Sec. B.3); with noise scale $\sigma = 0.5$ this gives $\tau^2 = 0.25$. The estimate is updated online using only the *completed* pairs $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})$ that are causally available at the extraction position $\boldsymbol{x}_t$ (the prediction location for $y_t$; cf. Sec. C.1); by convention $\hat{\boldsymbol{w}}_{\mathrm{ridge}}(0) = \boldsymbol{0}$ (the prior mean). $\boldsymbol{x}_t$ (current input); $\boldsymbol{x}_t^\top \hat{\boldsymbol{w}}_t$ (diagnostic prediction score); $\|\boldsymbol{x}_t\|_2^2$ (input norm).

We report held-out linear-probe $R^2$ (80/20 split) for each individual feature and for the combined set. To test the causal significance of the linearly accessible content, we extract the *filtered* subspace $\boldsymbol{V}_{\mathrm{filt}}$: fit a linear map from the combined features to $z$, take the SVD of the fitted values, and retain directions explaining $> 0.5\%$ of total variance. We then run the causal intervention with $\boldsymbol{V}_{\mathrm{filt}}$ in place of $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ and compare the resulting loss increases.

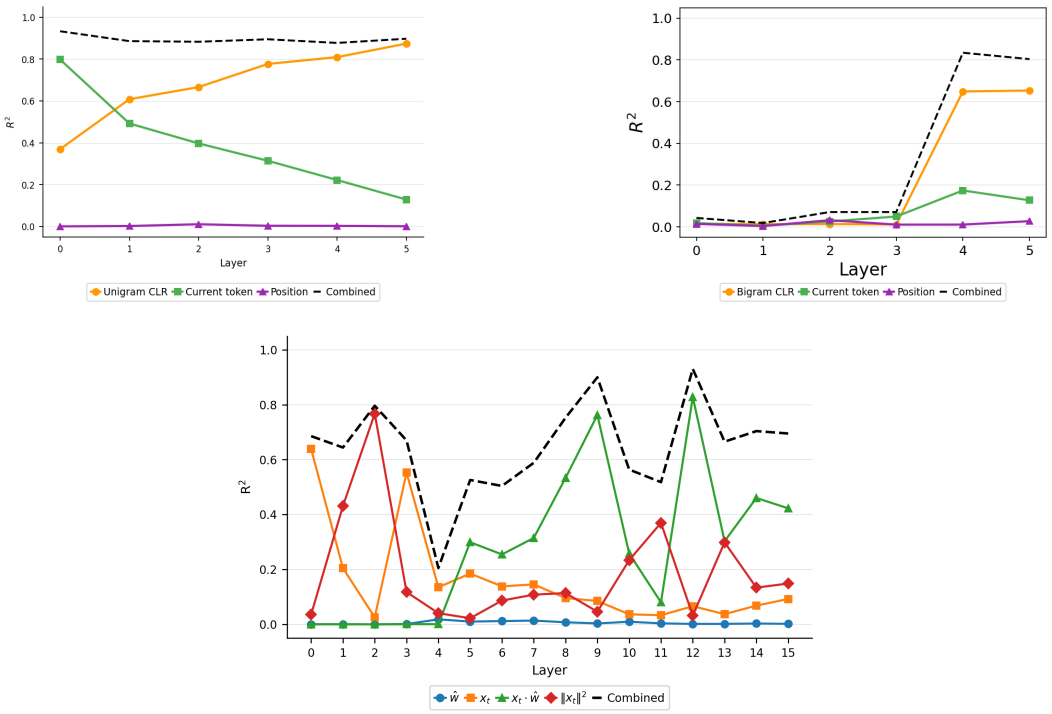



Figure 29: **Per-feature $R^2$ predicting $\hat{\boldsymbol{V}}_{\text{opt}}^{\top}\boldsymbol{h}$ across all three experiments.** Solid lines show individual feature contributions; the dashed line shows the combined $R^2$ from regressing on all features jointly (see Eq. 24 for the CLR definition used in E1 and E3). **(a) E1 (Dice):** current token dominates early ($R^2 \approx 0.81$ at layer 0) and decays, while unigram CLR rises from $\approx 0.35$ to $\approx 0.88$ by layer 5; combined $R^2$ remains 0.83–0.94 throughout. **(b) E3 (Latent Markov):** combined $R^2 \approx 0.01$–$0.04$ at layers 0–2, rising to $\approx 0.61$ at layer 3 and 0.79–0.85 at layers 4–5; bigram CLR becomes the dominant predictor ($R^2 \approx 0.75$ at layer 5). **(c) E2 (Linear Regression):** in early layers, the input statistics dominate, with $\boldsymbol{x}_t$ peaking at $R^2 \approx 0.64$ at layer 0 and $\|\boldsymbol{x}_t\|^2$ peaking at $R^2 \approx 0.77$ at layer 2; from layer 5 onward, the (causal) prediction score $\boldsymbol{x}_t \cdot \hat{\boldsymbol{w}}_t$ takes over and peaks at $R^2 \approx 0.76$ at layer 9 and $R^2 \approx 0.83$ at layer 12; $\hat{\boldsymbol{w}}_t$ stays near zero throughout; combined $R^2$ ranges from $\approx 0.21$ at the layer-4 transition dip up to $\approx 0.93$ at layer 12.

**Results: Dice (E1).** Figure 29(a) reveals that the orthogonal direction encodes different context statistics at different depths. In the early layers (0–2), the current token one-hot already explains the majority of $\hat{\boldsymbol{V}}_{\text{opt}}$ ($R^2 \approx 0.81$ at layer 0), while the unigram CLR contributes $R^2 \approx 0.35$; together the combined features achieve $R^2 \approx 0.94$. As depth increases, the current token's contribution decays to $\approx 0.14$ by layer 5, while the unigram CLR rises steadily to $\approx 0.88$. The combined $R^2$ remains high throughout (0.83–0.94), indicating that observable token statistics explain $\hat{\boldsymbol{V}}_{\text{opt}}$ well at every layer. Figure 30(a) confirms this causally: the filtered bars are nearly indistinguishable from $\hat{\boldsymbol{V}}_{\text{opt}}$ at every layer (≈75–135% for both OOD and minor), reproducing its causal effect almost exactly. This shows that what the orthogonal directions encode changes across layers, shifting from raw token identity in early layers to the running empirical frequency distribution in later layers, but the content is linearly accessible from observable statistics throughout.

**Results: Latent Markov (E3).** The Latent Markov setting presents a markedly different pattern (Figures 29(b) and 30(b)). The combined observable features explain virtually none of $\hat{\boldsymbol{V}}_{\text{opt}}$ in the first three layers ($R^2 \approx 0.01$–$0.04$ at layers 0–2), then rise sharply to $\approx 0.61$ at layer 3 and $\approx 0.79$–$0.85$ at layers 4–5. At layer 3, current token and bigram CLR contribute comparably ($R^2 \approx 0.34$ and 0.26 respectively); beyond that, bigram CLR becomes the dominant predictor ($R^2 \approx 0.75$ at layer 5) while current token decays to $\approx 0.10$ and position contributes negligibly throughout. The filtered intervention (Figure 30(b)) mirrors this pattern causally: at layer 1 the filtered bars are near zero (≈10%) despite $\hat{\boldsymbol{V}}_{\text{opt}}$ causing ≈115–123% disruption. Layers 0 and 2–3 show partial filtered effects (≈25–56% vs $\hat{\boldsymbol{V}}_{\text{opt}}$'s ≈70–102%), and at layers 4–5 the filtered bars closely track $\hat{\boldsymbol{V}}_{\text{opt}}$ (≈65–85%).

This reveals a genuine developmental transition: in early layers $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ encodes prediction-relevant information that is not linearly accessible from simple token statistics, whereas in later layers the orthogonal directions develop a linearly readable bigram-based predictor.

**Results: Linear Regression (E2).** Figures 29(c) and 30(c) show the results for E2. The $R^2$ plot (Figure 29(c)) reveals a two-phase per-feature progression separated by a sharp transition. In the early layers (0–3) the input statistics dominate: the current input $\boldsymbol{x}_t$ peaks at $R^2 \approx 0.64$ at layer 0 (with a secondary peak of $\approx 0.55$ at layer 3), and the input norm $\|\boldsymbol{x}_t\|^2$ peaks at $R^2 \approx 0.77$ at layer 2, indicating that the orthogonal direction in this regime tracks the raw current input together with its scale. From layer 5 onward, the prediction score $\boldsymbol{x}_t \cdot \hat{\boldsymbol{w}}_t$ (the inner product of the current input with the causal running ridge estimate, which measures how well the input aligns with the estimated task direction) takes over as the dominant feature, peaking at $R^2 \approx 0.76$ at layer 9 and $R^2 \approx 0.83$ at layer 12; $\|\boldsymbol{x}_t\|^2$ persists as a moderate secondary contributor in the deeper layers (with sporadic peaks of $\approx 0.23$–$0.37$ at layers 10–11 and 13). The running estimate $\hat{\boldsymbol{w}}_t$ itself contributes negligibly throughout, confirming that the compressed orthogonal direction tracks the scalar prediction score rather than the full ridge/weight estimate. The combined $R^2$ ranges from $\approx 0.21$ at the layer-4 transition dip to $\approx 0.93$ at layer 12, with a notable dip at layer 4 coinciding with the transition between the input-statistics-dominated and $\boldsymbol{x}_t \cdot \hat{\boldsymbol{w}}_t$-dominated regimes. The filtered intervention (Figure 30(c)) closely reproduces $\hat{\boldsymbol{V}}_{\mathrm{opt}}$'s causal effect on OOD and minor losses across most layers, with filtered and full bars essentially indistinguishable at every layer except layer 4. At the layer-4 transition, the combined probe $R^2$ dips to $\approx$0.21 and the $R^2 \geq 0.05$ filter retains only one of the two $V_{\mathrm{opt}}$ directions (with low joint $R^2$); the filtered bar consequently collapses to $\approx 12\%$ even though $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ itself disrupts $\approx 182\%$ of the ICL gain. This single-layer outlier coincides exactly with the input-statistics-to-prediction-score regime change identified in the $R^2$ analysis above, and the filtered subspace recovers $\hat{\boldsymbol{V}}_{\mathrm{opt}}$'s causal effect once the joint $R^2$ rises again at deeper layers, confirming that the causally active content elsewhere is well explained by these observable statistics.

**Summary.** The three settings reveal distinct patterns in both *what* and *when* $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ encodes prediction-relevant context. In E1 (Dice), simple token statistics explain $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ well at all depths ($R^2 \geq 0.83$ combined), but the dominant feature shifts across layers: current token identity predominates early while unigram CLR takes over in later layers. The filtered subspace accounts for essentially all of $\hat{\boldsymbol{V}}_{\mathrm{opt}}$'s causal power throughout the network. In E3 (Latent Markov), token statistics explain almost nothing in layers 0–2 ($R^2 \approx 0.01$–$0.04$), then become strongly predictive from layer 3 onwards as bigram CLR and current token emerge jointly ($R^2 \approx 0.61$ at layer 3, rising to 0.79–0.85 at layers 4–5). Correspondingly, the filtered intervention is near zero at layer 1 and partial at layers 0 and 2–3, only reproducing the full causal effect at layers 4–5; the large early-layer effects of $\hat{\boldsymbol{V}}_{\mathrm{opt}}$ are not linearly explainable by observable token statistics. In E2 (Linear Regression), the dominant feature shifts from input statistics in the early layers, where $\boldsymbol{x}_t$ peaks at $R^2 \approx 0.64$ at layer 0 and $\|\boldsymbol{x}_t\|^2$ peaks at $R^2 \approx 0.77$ at layer 2, to the (causal) prediction score $\boldsymbol{x}_t \cdot \hat{\boldsymbol{w}}_t$ from layer 5 onward (peak $R^2 \approx 0.83$ at layer 12), with $\hat{\boldsymbol{w}}_t$ contributing negligibly throughout. The filtered subspace accounts for the causal effect on OOD and minor tasks across most layers, with the exception of the layer-4 $R^2$ transition dip where the filter discards the causally active directions. In all cases, the filtered subspace $\boldsymbol{V}_{\mathrm{filt}}$ reproduces essentially all of $\hat{\boldsymbol{V}}_{\mathrm{opt}}$'s causal power once the combined $R^2$ is appreciable, directly confirming that $\mathrm{col}(\boldsymbol{U}_{\perp})$ stores linearly accessible running prediction statistics that drive extrapolative in-context learning.

### D.11 Near-orthogonality of the OOD subspace is learned: principal angles against a random-subspace null

Empirically, OOD representations in the M2 regime are near-orthogonal to the ID task subspace (Sec. 6.1). In a hidden dimension of $d = 128$ this could arise by chance, since two random low-dimensional subspaces are already nearly orthogonal. This section shows that the observed near-orthogonality is not inherited from random initialization but is learned during training. Early in training the loss-relevant OOD direction overlaps the task subspace far above chance, and it moves to chance-level overlap as M2 develops. Unlike a random direction, the learned direction is also predictive of the OOD statistics and necessary for OOD prediction.

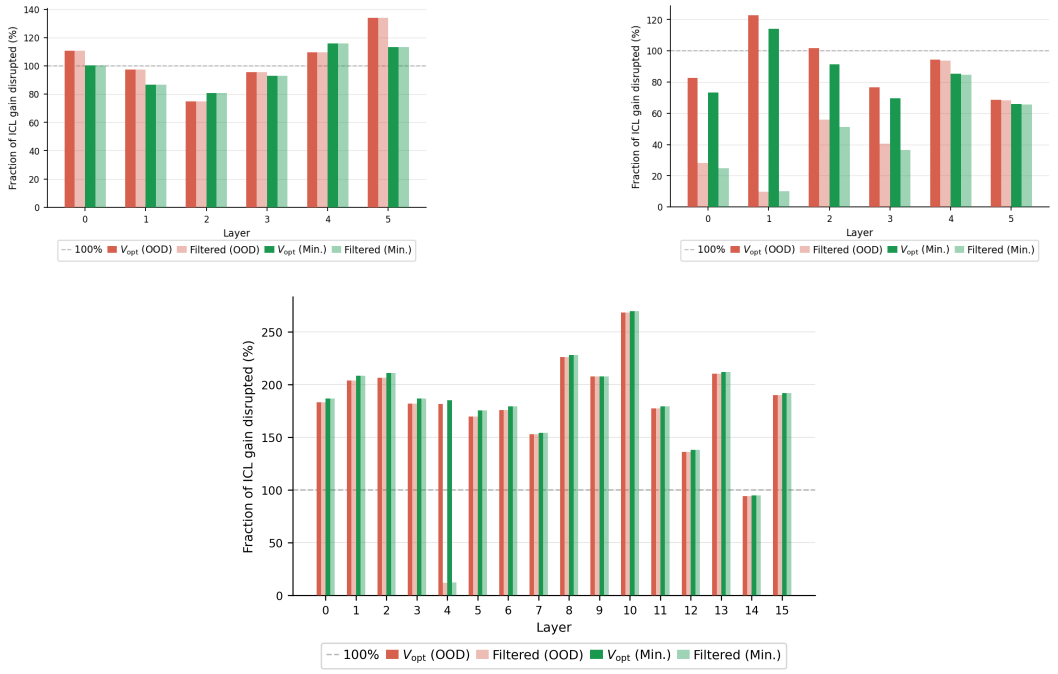



Figure 30: **Causal comparison of $\hat{\boldsymbol{V}}_{\text{opt}}$ vs. filtered subspace $\boldsymbol{V}_{\text{filt}}$ across all three experiments.** Dark bars show the full $\hat{\boldsymbol{V}}_{\text{opt}}$ intervention; light bars show the filtered subspace derived from the combined observable features. **(a) `E1` (Dice):** filtered bars are nearly indistinguishable from $\hat{\boldsymbol{V}}_{\text{opt}}$ at every layer ($\approx$75–135%), confirming that the causally active content is linearly accessible from token statistics throughout. **(b) `E3` (Latent Markov):** filtered bars are near zero at layer 1 ($\approx$10%) despite $\hat{\boldsymbol{V}}_{\text{opt}}$ causing $\approx$115–123% disruption; partial at layers 0 and 2–3 ($\approx$25–56%); closely match $\hat{\boldsymbol{V}}_{\text{opt}}$ at layers 4–5 ($\approx$65–85%), confirming a developmental transition in linear accessibility. **(c) `E2` (Linear Regression):** filtered bars closely match $\hat{\boldsymbol{V}}_{\text{opt}}$ at every layer except layer 4; the layer-4 filtered bar collapses to $\approx$12% because the combined probe $R^2$ dips to $\approx$0.21 at the input-statistics-to-prediction-score transition and the $R^2 \geq 0.05$ filter retains a direction with low joint explanatory power, missing the causally active component there.

**Setup.** We use the `E3` model at diversity exponent $m = 10$, where the model is in the `M2` mode, and repeat the analysis at $m = 6$. Our analysis covers every layer. The task subspace $\mathrm{col}(\hat{\boldsymbol{\Theta}})$ is estimated by averaging task vectors over sequences that are not used anywhere else in the analysis.

**Two estimates of the OOD subspace.** The probe subspace is the row space of a ridge regression of OOD hidden states on the bigram transition statistics of Sec. D.10. It captures where the in-context information is linearly decodable. The loss-relevant subspace consists of the directions found by gradient ascent on the OOD loss with no constraint on where they lie. It captures the directions that matter for OOD prediction.

**Affinity and the random-subspace null.** We measure how much an OOD subspace $\boldsymbol{U}$ overlaps the task subspace by its affinity $\rho$, defined as the mean squared cosine of the principal angles between $\boldsymbol{U}$ and $\mathrm{col}(\hat{\boldsymbol{\Theta}})$. Equivalently, $\rho$ is the fraction of the energy of $\boldsymbol{U}$ that lies inside the task subspace. It equals one when $\boldsymbol{U}$ lies inside the task subspace and zero when the two are orthogonal. Task subspaces estimated from two disjoint halves of the held-out sequences have affinity 0.92–0.999 to each other, so the estimation error of $\mathrm{col}(\hat{\boldsymbol{\Theta}})$ is small. The null distribution is that of $\rho$ for 2,000 random subspaces of the same dimension as $\boldsymbol{U}$, drawn from the Haar measure. Its mean is $r/d$ for a rank-$r$ subspace. At $d = 128$ this is about 0.016 for the rank-2 loss-relevant subspace and 0.05–0.08 for the probe subspace of rank 7–10. Random subspaces are therefore nearly but not exactly orthogonal to the task subspace. An angle alone cannot separate a learned subspace from a random one, but the comparison to the matched null and the behavior across training can.

Table 12: **Affinity of the loss-relevant OOD subspace to the task frame, per layer (final checkpoint).** Affinity $\rho$ of the loss-relevant subspace at $m = 10$ ($m = 6$ in parentheses), the 95% band of the matched rank-2 Haar null, the position relative to that band, and the affinity of the probe subspace.

| layer | loss-relevant $\boldsymbol{U}$ $\rho$ | Haar null 95% band | relative to null | probe $\boldsymbol{U}$ $\rho$ |
|---|---|---|---|---|
| 0 | 0.145 (0.036) | $[0.002, 0.043]$ | above null ($m = 10$; $p < 0.002$) | 0.912 (0.803) |
| 1 | 0.079 (0.042) | $[0.002, 0.043]$ | above null ($m = 10$) | 0.849 (0.795) |
| 2 | 0.0039 (0.0038) | $[0.002, 0.043]$ | within null band ($p = 0.16$ / 0.17) | 0.995 (0.919) |
| 3 | 0.0135 (0.0045) | $[0.002, 0.044]$ | within null band ($p = 0.97$ / 0.22) | 0.837 (0.485) |
| 4 | 0.0147 (0.0155) | $[0.002, 0.043]$ | within null band | 0.037 (0.279) |
| 5 | 0.0245 (0.0184) | $[0.002, 0.043]$ | within null band | 0.083 (0.159) |

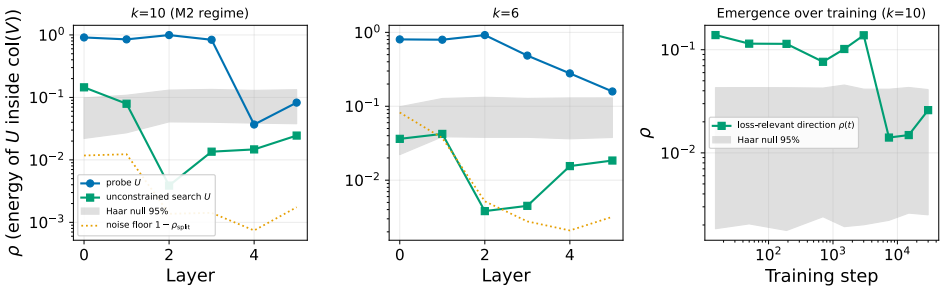



Figure 31: **Principal-angle affinity against the Haar null. Left, center:** per-layer affinity $\rho$ of the probe and loss-relevant OOD subspaces to the task frame at $m = 10$ and $m = 6$ (log scale), with the 95% band of the matched Haar null (gray) and the estimation noise floor (dotted). **Right:** affinity of the loss-relevant direction at the band layer across training checkpoints ($m = 10$): it starts at 5–9$\times$ the null and descends into the null band over the window in which M2 develops.

**Results.** Table 12 and Fig. 31 give three results. First, at layers 2–5, which include the analysis band, the loss-relevant OOD direction is as orthogonal to the task subspace as a random direction. Its affinity $\rho = 0.004$–$0.025$ lies inside the null band. At layers 0–1 it lies far above the null ($\rho = 0.08$–$0.15$, $p < 0.002$ at $m = 10$). These are the layers of shared computation in Sec. D.8, where task and prediction information are not yet separated.

Second, the orthogonality is learned. We repeat the gradient ascent across training checkpoints at the band layer. The loss-relevant direction starts at 5–9 times the null overlap, with $\rho = 0.08$–$0.14$ and largest cosine up to 0.50 between steps 15 and 3000. It then drops into the null band ($\rho = 0.014$–$0.026$) over the same training window in which M2 develops (Fig. 5). If the orthogonality came from initialization, this trace would be flat.

Third, the component of OOD representations that does lie in the task subspace is informative. The OOD transition statistic predicts the task-subspace coordinates of OOD hidden states with $R^2 = 0.49$, against 0.05 for random directions orthogonal to the task subspace. The model thus represents inside the simplex which major tasks the OOD task resembles. This is why the probe subspace overlaps the task subspace substantially at layers 0–3, while the direction that matters for the loss is orthogonal to it. Together with the causal evidence of Sec. E, where removing the learned direction costs up to 0.28 nats of OOD loss and removing random orthogonal directions costs about 0, the learned subspace has four properties. It is orthogonal to the task subspace at chance level, predictive, causally necessary, and emergent under training. A random direction has only the first.

**Why orthogonal?** We offer a tentative explanation for why the model uses near-orthogonal subspaces to represent the two modes. First, from classical results on packing feature directions into a limited number of dimensions, we know that the arrangement with the least interference is orthogonal when the number of features is at most the dimension and a tight frame when it exceeds the dimension [57, 10], and neural networks have been observed to store features this way [16]. These results motivate the picture but do not directly apply to transformers. Second, a heuristic mechanism points the same way. Downstream layers read the task subspace as a posterior over training tasks (Theorem 1), so any OOD component inside $\mathrm{col}(\hat{\boldsymbol{\Theta}})$ would corrupt the in-distribution predictions, and the training loss pushes that component into the orthogonal complement. Third, the checkpoint

results above agree with this mechanism, since the orthogonality is absent early in training and forms over the window in which the extrapolative mode develops. An orthogonality theorem would likely require stronger assumptions than Theorem 2, and we leave it to future work.

### D.12 Layer-band selection and a replication of Table 2 with fixed bands

Table 2 reports each intervention averaged over a band of mid-depth layers (the simplex injection uses one layer). This section explains how the band is chosen, offers a tentative explanation for why the task-subspace $R^2$ peaks at mid-depth, and repeats the experiment of Table 2 with the band fixed in advance and new seeds.

**Remark 5** (Depth dependence)**.** *The empirical claims in this paper refer to a mid-depth band of layers, selected as the layers maximizing held-out ID task-subspace $R^2$ (see Sec. D.12), because the properties are not uniform across depth.* `P0` *holds at every layer at long context (Table 1a) and* `P3` *strengthens with depth up to the middle layers (Sec. C.4.3), but* `P1` *and* `P2` *fail at the last layers of* `E2` *and* `E3`*, so a task vector extracted there satisfies none of* `P1`*–*`P3`*. Task information likely persists at the last layers but is re-encoded for next-token prediction rather than kept as an additive simplex; see Sec. C.3 for the supporting probe decomposition and Sec. D.12 for a tentative explanation.*

**Selection criterion.** The band is the set of layers around the layer with the highest held-out ID task-subspace $R^2$ (the projection $R^2$ of Sec. 6.1 on ID sequences). The criterion uses only the task-vector geometry and no intervention. In the synthetic models it agrees with two other indicators. The task–token interaction $\eta^2_{\text{interaction}}$ is small at these layers (Table 1(b)), and the per-layer intervention effects are mode-selective there (Sec. D.8). The same criterion selects the layers analyzed in the pretrained models (Sec. D.5–D.6).

**Why a mid-depth band: a tentative explanation.** We offer a tentative explanation for why the task-subspace $R^2$ peaks at mid-depth. In the early layers the model computes statistics shared by both modes. At layers $0$–$1$ the loss-relevant OOD subspace overlaps the task subspace far above chance (Sec. D.11), and the unique task contribution in the probe is small (Sec. C.3). In the middle layers the sufficient statistic of the context is complete, and hidden states lie close to the task subspace. The task–token interaction is small (Table 1b), the unique task contribution peaks, and interventions are mode-selective (Sec. D.8). In the last layers the hidden state is re-encoded for next-token prediction. The interaction share rises, the model's own prediction accounts for $0.86$ of the residual stream, and interventions lose selectivity because both modes feed the same prediction.

Two further observations agree with this picture. When the sufficient statistic takes more computation, as with order-3 data, the band moves one layer deeper (Sec. E). In the pretrained models the same criterion selects layers at $41$–$71\%$ relative depth (Sec. D.6). The three-stage picture also matches stages-of-inference analyses of transformer depth [49, 30, 3, 21], the finding that intermediate layers carry the strongest linear features across architectures [44], and the common practice of extracting task and function vectors [20, 51] and of editing and steering [26, 41] at middle layers. Late layers still contain task information, but no longer as a separate direction that a linear projection recovers, and we leave the characterization of the late-layer re-encoding to future work.

**Replication with the bands fixed in advance.** The band was originally identified partly from the per-layer intervention profile. To check that the effects in Table 2 do not depend on this choice, we repeated the experiment with the bands fixed at their published values and all data redrawn under three new seeds, for `E1` (layers 1–4, diversity exponent $m = 10$) and `E3` (layers 2–4, $m = 12$). Table 13 reproduces the double dissociation. Suppressing $\mathrm{col}(\hat{\boldsymbol{\Theta}})$ erases more than the full in-context gain on major tasks while leaving OOD loss largely intact, and suppressing $\hat{\boldsymbol{V}}_{\text{opt}}$ erases a large fraction of the OOD gain while the random same-rank control stays near $3\%$. The magnitudes vary with the seed and across the layers of the band, but in every seed and both experiments the two interventions dissociate, which is the claim of Table 2.

## E Beyond First-Order Markov: Finite-Order Sweep and the Dyck Boundary

The main text studies first-order Markov sources (`E1`–`E3`) and the Dyck language (`E4`), which has no finite order. This section covers the range between them. We repeat the main analyses of

Table 13: **Replication of Table 2 with the layer bands fixed in advance.** Relative loss degradation $\Delta\mathcal{L}_{\text{mode}}/g_{\text{mode}} \times 100\%$, band mean $\pm$ sd over three new seeds, with the values of Table 2 in parentheses. "Rand." is the random same-rank control of the orthogonal intervention.

| | | Task Subspace Interv. (%) | | Orth. Subspace Interv. (%) | | |
|---|---|---|---|---|---|---|
| Exp. | Band | Maj. | OOD | Maj. | OOD | Rand. |
| E1 | 1–4 | $177.0 \pm 8.2$ (211.1) | $31.7 \pm 0.4$ (12.1) | $34.3 \pm 13.4$ (4.0) | $131.0 \pm 6.4$ (95.8) | $3.4 \pm 0.1$ (2.9) |
| E3 | 2–4 | $129.7 \pm 23.1$ (132.0) | $-1.6 \pm 0.1$ (13.1) | $3.8 \pm 0.6$ (2.1) | $71.6 \pm 5.8$ (99.6) | $3.3 \pm 0.2$ (3.0) |

Secs. 5–6 on mixtures of order-$w$ Markov chains for $w$ up to 10, and then measure why no compact belief state forms for Dyck. Throughout, $m$ is the diversity exponent and the mode diagnostic is $\log(\mathrm{KL}_{\text{Bayesian}}/\mathrm{KL}_{\text{extrapolative}})$ of Sec. 6 at the final checkpoint, negative for M1 and positive for M2. The value of $m$ at which it changes sign, $m^*$, is the transition point between the two modes.

**Setup.** An order-$w$ chain conditions on the last $w$ tokens, $\mathbb{P}(s_{t+1} \mid z, s_{\le t}) = \mathbb{P}(s_{t+1} \mid z, s_{t-w+1:t})$. The latent task $z$ is a transition tensor with $V^w$ rows, each drawn from $\mathrm{Dir}(\mathbf{1}_V)$ as in E3, and $w = 1$ recovers E3. The extrapolative predictor M2 is the smoothed empirical $(w{+}1)$-gram, and M1 is the exact posterior over the training pool under the order-$w$ likelihood. The sweep uses vocabulary $V = 8$, the training mixture of E3, and the same 6-layer, $d = 128$ model. A sequence of $T = 192$ tokens provides fewer than 200 transitions while an order-$w$ chain has $V^w$ contexts, so at $V = 8$ the chain cannot be estimated from one sequence beyond $w = 3$. To reach order 10 we also use $V = 2$. All models are trained for 30k steps; the $V = 8$ models of orders 1–3 are also trained for 100k steps, and we report those where available.

**Finite order within the framework.** An order-$w$ chain is a first-order chain on the space $\mathcal{V}^w$ of windows of $w$ tokens. The framework of Sec. 4 therefore applies once the hidden state is conditioned on the task and the last $w$ tokens, $(z, s_{t-w+1:t})$, instead of the task and the current token, $(z, s_t)$, as in Eq. 1. We call the number of most recent tokens that an analysis conditions on its conditioning window, $w_{\text{cond}}$; the main text uses $w_{\text{cond}} = 1$. Dyck lies outside this family, since the next planted bracket depends on the whole bracket prefix.

Table 14: **Mode diagnostic versus Markov order ($V = 8$).** Final $\log(\mathrm{KL}_{\text{Bayesian}}/\mathrm{KL}_{\text{extrapolative}})$ as a function of the diversity exponent $m$; negative favors M1. Orders 1–3 are from the 100k-step grid; at $w \ge 4$ (shown at 30k steps) neither reference predictor can be computed from one context, the ratio is about $+0.15$ for every $m$, and no transition point is defined.

| order $w$ | steps | $m$=0 | $m$=2 | $m$=4 | $m$=6 | $m$=8 | $m$=10 | transition $m^*$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 100k | $-5.53$ | $-5.34$ | $-4.64$ | $-3.29$ | $-0.75$ | $+2.07$ | $\approx 8.5$ |
| 2 | 100k | $-4.69$ | $-4.30$ | $-3.35$ | $-0.28$ | $+3.61$ | $+4.65$ | $\approx 6.1$ |
| 3 | 100k | $-3.71$ | $-2.44$ | $+1.30$ | $+2.78$ | $+3.45$ | $+3.43$ | $\approx 3.3$ |
| 4 | 30k | $+0.12$ | $+0.18$ | $+0.20$ | $+0.21$ | $+0.19$ | $+0.16$ | – |
| 5 | 30k | $+0.14$ | $+0.16$ | $+0.16$ | $+0.16$ | $+0.16$ | $+0.14$ | – |

Table 15: **Binary-vocabulary ladder to order 10 ($V = 2$, 30k steps).** Same diagnostic as Table 14. With $V = 2$ the context is long enough to estimate the chain at every order shown; the transition is present at order 10 and its transition point moves to lower diversity as $w$ grows.

| order $w$ | $m$=0 | $m$=2 | $m$=4 | $m$=6 | $m$=8 | $m$=10 | transition $m^*$ |
|---|---|---|---|---|---|---|---|
| 1 | $-0.67$ | $-0.16$ | $-0.16$ | $-0.06$ | $-0.02$ | $+0.00$ | $\approx 9.7$ |
| 3 | $-0.93$ | $-1.09$ | $-0.86$ | $-0.34$ | $+0.31$ | $+0.38$ | $\approx 7.0$ |
| 5 | $-2.20$ | $-1.97$ | $-1.36$ | $-0.28$ | $+1.70$ | $+1.62$ | $\approx 6.3$ |
| 7 | $-3.39$ | $-2.74$ | $-1.39$ | $+0.94$ | $+1.92$ | $+2.60$ | $\approx 5.2$ |
| 10 | $-2.12$ | $-0.58$ | $+1.65$ | $+2.24$ | $+2.56$ | $+2.45$ | $\approx 2.5$ |

**The transition at every finite order.** Table 14 and Fig. 32 show that the transition of Fig. 5 appears at every order at which both reference predictors can be computed from one context. The transition

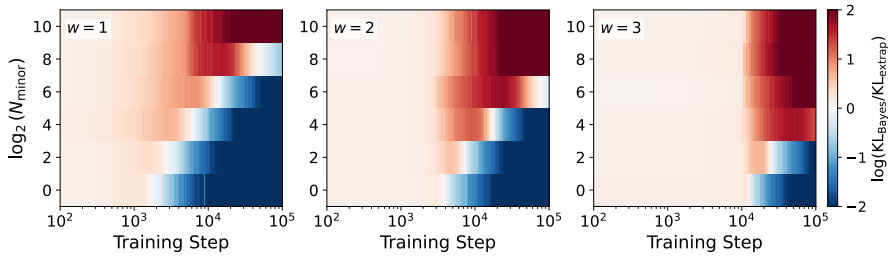



Figure 32: **Phase transition at orders $w = 1, 2, 3$ ($V = 8$, 100k steps).** Same layout and color scale as Fig. 5: training step on the horizontal axis, diversity exponent $m = \log_2 N_{\text{minor}}$ on the vertical axis; blue indicates `M1`, red `M2`. The diagonal boundary is present at every order and moves down (toward lower diversity) as $w$ increases.

point moves to lower diversity as $w$ grows ($m^* \approx 8.5, 6.1, 3.3$ at $w = 1, 2, 3$). This is the direction Theorem 2 predicts, since the order-$w$ family has intrinsic dimension $V^w(V-1)$ and the extrapolative predictor therefore leaves a fixed task subspace at a smaller $N_{\text{minor}}$. At $w \geq 4$ with $V = 8$ neither predictor can be computed from one context, and the diagnostic is about $+0.15$ at every $m$. With $V = 2$ the transition is present up to order 10 and its transition point again moves to lower diversity with $w$ (Table 15), so the limit at $V = 8$ comes from context length rather than from order.

**Training length.** Between 30k and 100k steps the transition point at $V = 8$ shifts to higher diversity by about 1.2 at every order, so longer training slightly favors retrieval, in line with Carroll et al. [9]. The transition and its ordering across $w$ are unchanged.

Table 16: **Variance decomposition with a wider conditioning window at layer 3, $m = 6$ (30k steps).** Residual variance ratio $\text{SS}_{\text{within}}/\text{SS}_{\text{total}}$ (lower is better) at a middle and at the last context position, and the interaction share $\eta^2_{\text{int}}$ at the last position, when conditioning on the current token only ($w_{\text{cond}} = 1$) and on the last $w$ tokens ($w_{\text{cond}} = w$). Left: $V = 8$; right: $V = 2$.

| $V = 8$ | | | | | $V = 2$ ladder | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $w$ | $w_{\text{cond}}$ | resid. (mid) | resid. (last) | $\eta^2_{\text{int}}$ (last) | $w$ | $w_{\text{cond}}$ | resid. (mid) | resid. (last) | $\eta^2_{\text{int}}$ (last) |
| 1 | 1 | 0.360 | 0.160 | 0.254 | 1 | 1 | 0.254 | 0.181 | 0.235 |
| 2 | 1 | 0.702 | 0.664 | 0.096 | 3 | 1 | 0.643 | 0.478 | 0.340 |
| 2 | 2 | 0.220 | 0.130 | 0.010 | 3 | 3 | 0.361 | 0.182 | 0.202 |
| 3 | 1 | 0.800 | 0.794 | 0.198 | 5 | 1 | 0.761 | 0.653 | 0.285 |
| 3 | 3 | 0.065 | 0.059 | 0.004 | 5 | 5 | 0.346 | 0.168 | 0.203 |
| 4 | 1 | 0.780 | 0.780 | 0.178 | 7 | 1 | 0.861 | 0.831 | 0.236 |
| 4 | 4 | 0.059 | 0.057 | 0.004 | 7 | 7 | 0.187 | 0.157 | 0.030 |
| 5 | 1 | 0.771 | 0.773 | 0.165 | 10 | 1 | 0.905 | 0.899 | 0.235 |
| 5 | 5 | 0.019 | 0.019 | 0.001 | 10 | 10 | 0.084 | 0.081 | 0.005 |

**`P0`–`P1`: the conditioning window must match the order.** Table 16 applies the variance decomposition of Sec. 5.1 to the order-$w$ models. When the analysis conditions on the current token only, as in the main text, the residual variance ratio grows with order, reaching 0.77 at $w = 5$ for $V = 8$ and 0.90 at $w = 10$ for $V = 2$. Conditioning on the last $w$ tokens instead restores additivity at every order with $V = 8$ and at orders 7 and 10 with $V = 2$. The residual falls to 0.02–0.08, with interaction share at most 0.005. At orders 3 and 5 with $V = 2$ the residual falls to 0.17–0.18 but an interaction share of 0.20 remains. The 100k models are better still (Fig. 33). `P0` and `P1` therefore hold once the conditioning window matches the order, and what fails at higher order is mainly conditioning on a single token.

**`P3` at every order.** Table 17 repeats the analysis of Fig. 3 with the order-aware posterior, using the Hellinger distance of Eq. 23. The simplex coefficients track the posterior equally well at every order, far from the uniform reference, and OOD sequences stay far from any major-only posterior. The only change with order is depth. At $w = 3$ the best layer is one layer deeper, since the 3-gram statistic takes more computation.

**Orthogonal interventions at every order.** We run two versions of the orthogonal intervention of Sec. 6.3 on the $m = 10$ models at each order. The first removes a single learned direction orthogonal

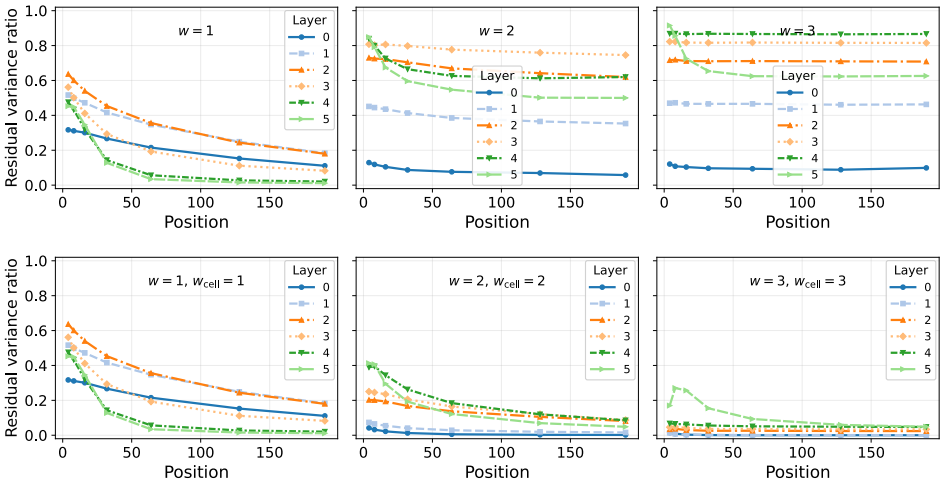



Figure 33: **Residual variance ratio across context positions and layers ($V = 8$, 100k steps, $m = 6$).** Panels: $w = 1, 2, 3$. **Top:** conditioning on the current token only ($w_{\text{cond}} = 1$); at $w = 2, 3$ most of the variance is left unexplained at every position. **Bottom:** conditioning on the last $w$ tokens ($w_{\text{cond}} = w$); the curves return to the E3 profile of Table 1(a), decaying with position and with depth.

Table 17: **Posterior alignment versus order ($V = 8$).** Hellinger distance in late context between the simplex coefficients $\beta_{t,k}$ and the order-aware Bayesian posterior $\alpha_{t,k}$ on ID sequences; the OOD column compares against the major-only posterior and the last column is a uniform-random reference. At order 3 the best layer is one layer deeper.

| order $w$ | layer | ID vs posterior | OOD vs major-only posterior | uniform-random |
|---|---|---|---|---|
| 1 | 3 | 0.116 | 0.575 | 0.660 |
| 2 | 3 | 0.129 | 0.570 | 0.660 |
| 3 | 4 | 0.127 | 0.562 | 0.660 |

Table 18: **Rank-1 orthogonal-direction intervention per order ($m = 10$).** Change in cross-entropy (nats) when the single direction, orthogonal to both the task and token subspaces, whose removal most damages OOD prediction is projected out, compared with random orthogonal directions.

| order $w$ | max $\Delta$ OOD (layer) | $\Delta$ ID at that layer | mean $\Delta$ ID over layers | random direction |
|---|---|---|---|---|
| 1 | $+0.031$ (L0) | $+0.003$ | $+0.001$ | $+0.0001$ |
| 2 | $+0.043$ (L4) | $-0.000$ | $+0.002$ | $-0.0000$ |
| 3 | $+0.276$ (L5) | $+0.002$ | $+0.019$ | $+0.0000$ |

Table 19: **The intervention of Table 2 per order ($m = 10$, layers 2–5).** Rank-6 subspace found by optimization on held-out minor-task data, $\gamma = 2.5$, post-MLP, with random same-rank controls (Sec. D.9). Each entry reports $\Delta$OOD loss in nats / $\Delta$major loss as a percentage of the major baseline / $\Delta$OOD loss of the random control in nats.

| layer | order 1 | order 2 | order 3 |
|---|---|---|---|
| 2 | $+0.16$ / 0.3% / $+0.006$ | $+0.28$ / 7.3% / $-0.003$ | $+0.31$ / 7.4% / $+0.000$ |
| 3 | $+0.20$ / 0.7% / $+0.008$ | $+0.34$ / 6.4% / $+0.016$ | $+0.40$ / 23.2% / $+0.025$ |
| 4 | $+0.44$ / 1.4% / $+0.020$ | $+0.31$ / 4.2% / $-0.001$ | $+0.49$ / 9.4% / $-0.002$ |
| 5 | $+0.26$ / 6.5% / $+0.004$ | $+0.50$ / 5.8% / $+0.010$ | $+1.49$ / 5.8% / $+0.024$ |

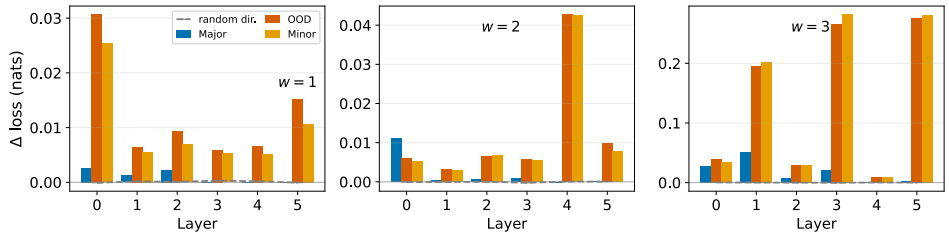



Figure 34: **Per-layer loss change under the rank-1 orthogonal-direction intervention of Table 18.** Panels: $w = 1, 2, 3$ (note the different vertical scales). Bars show $\Delta$ loss in nats for major, OOD, and minor sequences; the thin gray bars are random orthogonal directions. The dissociation sharpens with order and moves one layer deeper at $w = 3$.

to both the task and token subspaces (Table 18, Fig. 34). It costs up to $+0.28$ nats of OOD loss at order 3 while ID loss barely changes, and random orthogonal directions cost about 0. The dissociation holds at every order and grows with it. The second repeats the setting of Table 2 (Table 19). We report nats rather than the fraction of the in-context gain, because at orders 2–3 the OOD loss stays near the marginal entropy $\log 8$ throughout the context, so the gain is nearly zero and the fraction is undefined. At order 1 the fraction is well defined. Layers 2–4 erase 51–145% of the OOD gain and 3–12% of the major-task gain, in the range of the E3 row of Table 2. At order 3 the selective layer is layer 5, one deeper than at order 1, matching the depth shift seen for P3.

Table 20: **Conditional variance of Dyck hidden states (E4).** Fraction of hidden-state variance left unexplained after conditioning on the open-parenthesis count versus on the full bracket prefix, averaged over Dyck prefix lengths $l = 7$–18.

| layer | resid. fraction \| count | resid. fraction \| full prefix |
|---|---|---|
| 0 | 0.84 | 0.02 |
| 1 | 0.81 | 0.03 |
| 2 | 0.73 | 0.03 |
| 3 | 0.58 | 0.03 |
| 4 | 0.66 | 0.03 |
| 5 | 0.63 | 0.04 |

**The Dyck boundary: no compact belief state forms.** For Dyck, the natural candidate for a compact belief state is the running count of unmatched open parentheses. We ask whether the E4 model of Sec. 7 compresses its hidden state onto this count. Using 64 tasks, 8 planting masks with 32 samples each, and hidden states at every planted-bracket position, we compute the fraction of variance left after conditioning on the count and after conditioning on the full bracket prefix, averaged over prefix lengths 7–18. Table 20 shows that the count explains at most 42% of the variance, while the full prefix explains about 97% at every layer. This holds at every prefix length in the range. Averaged over layers, the residual given the count stays between 0.52 and 0.79, while the residual given the prefix stays at 0.02–0.04. The hidden state thus retains the prefix rather than a compact summary of it. This is the mechanism behind the failure of P0 in Sec. 7 and the prefix memorization in Sec. F.

In sum, the framework holds at every finite order once the conditioning window matches the order, and fails only for Dyck, which has no finite order.

## F Non-Markovian Case: Dyck Path Experiment (E4) Details

This section provides the methodology and detailed results behind the planted Dyck experiment (E4) used as the motivating non-Markovian counterexample in Sec. 7. The experiment matters for our main thesis because it shows that property P0 (long-context stability) is conditional on the data-generating structure rather than an innate property of trained transformers: when the source is non-Markovian and admits no compact sufficient statistic, the model retains the full prefix history instead of collapsing onto a low-dimensional summary. The main text (Figure 8) reported the residual variance left in $\boldsymbol{h}_t$ after conditioning on the task $z$ and a local context window, and visualized the

cluster structure of the final-layer hidden states at a single prefix length. This section sharpens that diagnosis along two complementary axes. First, we train linear probes that classify hidden states by the identity of the *full* planted Dyck prefix at every length $l = 2, \ldots, 7$ and report the resulting validation accuracies, providing direct quantitative evidence that the model retains an essentially lossless record of the prefix history rather than a compact summary statistic. Second, we show that this record is recoverable through a single shared low-dimensional projection across all prefix lengths and mask realizations, indicating that the prefix information is arranged along consistent directions in hidden-state space. Together these results constitute a sharp failure of property `P0` in `E4` that does not occur in `E1`–`E3`.

**Sequence construction.** The main text introduces `E4` (Sec. 7) by describing each sequence as a length-$T$ background of i.i.d. uniform tokens into which the latent Dyck string $z$ of length $2L$ is planted at positions selected with probability $\rho$. We expand on the planting rule here so the case $\rho T > 2L$ is unambiguous. Concretely, given $z = (z_1, \ldots, z_{2L})$ we draw a Bernoulli planting indicator vector $\boldsymbol{\xi} \in \{0,1\}^T$ with $\xi_t \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\rho)$, then enumerate its successes $1 \le \tau_1 < \tau_2 < \cdots$ in left-to-right order and overwrite background token $b_{\tau_j}$ with $z_j$ for every $j \in \{1, \ldots, \min(2L, \sum_{t=1}^{T} \xi_t)\}$. With the experimental settings used throughout this paper, $T = 192$, $2L = 20$, $\rho = 0.25$, the expected number of Bernoulli successes is $\rho T = 48 \gg 2L$, so undersampling (fewer than $2L$ successes) occurs with vanishingly small probability and the entire latent string $z$ is essentially always planted; the suffix of the sequence after the $2L$-th plant is pure background and therefore carries no further task-conditional signal.

**Prefix probe.** Consider the model trained on experiment `E4` with $K$ Dyck tasks (balanced bracket strings of length $2L = 20$). At each sequence position, define the *Dyck prefix length* $l$ as the number of planted bracket characters observed so far. Two tasks that share the same first $l$ planted characters are indistinguishable given the observed Dyck prefix up to that point. We group tasks by their length-$l$ prefix and define the *prefix class* of a hidden state as the identity of this prefix. The number of distinct prefix classes grows with $l$: for instance, $l = 2$ yields only 2 classes (open or close), while $l = 7$ yields 35 classes.

We train a linear probe consisting of a shared 2-dimensional linear projection $\boldsymbol{W} \in \mathbb{R}^{2 \times d}$ followed by a separate two-layer `MLP` classifier per prefix length. The probe is trained jointly across all prefix lengths $l = 2, \ldots, 7$ (at $l = 1$ there is a single class), using hidden states extracted from layer 5 (the final layer). Training and validation sets are constructed by generating sequences under multiple random planting masks and collecting hidden states at positions where exactly $l$ Dyck characters have been planted.

**Probing results.** The probe achieves near-perfect validation accuracy ($> 95\%$) for all prefix lengths up to $l = 7$ (the longest tested), reaching $98.6\%$ at $l = 7$ despite the rapid growth in the number of classes. Figure 35 shows a shared 2-dimensional linear projection at $l = 7$, where the 35 distinct prefix classes form well-separated clusters. Points are colored by the number of unmatched left parentheses, and the shaded regions depict the decision boundaries of the per-length `MLP` classifier. This visualization is identical to the right panel of Figure 8 in projection and coloring; the only difference is that here each cluster is labeled by its literal bracket prefix, whereas in the main-text panel the same prefixes are encoded compactly in hexadecimal for legibility. The hexadecimal encoding is obtained by mapping each step to a bit ('(' $\to 1$, ')' $\to 0$), dropping the leading '(' (present for all prefixes), padding the remaining bits to a multiple of 4, and converting to hexadecimal; for example, the length-7 prefix (()()( yields bits $1, 0, 1, 0, 0, 1$ after dropping the first '(', and padding to 8 bits gives $10100100_2 =$ `A4`.

This result has two implications. First, the hidden states encode a representation of the full Dyck prefix, not a compressed statistic such as the running difference between opening and closing brackets (which would yield only $\lfloor l/2 \rfloor + 1$ distinguishable states at prefix length $l$, not the valid Dyck-prefix classes observed). Second, because this encoding is recoverable through a shared 2-dimensional linear projection, the prefix information is arranged along consistent directions across different prefix lengths and mask realizations.

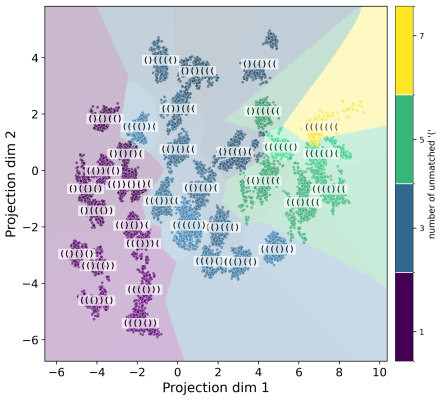



Figure 35: **Hidden-state geometry for the planted Dyck language (E4).** Two-dimensional projection of final-layer hidden states at prefix length $l = 7$ (yielding 35 distinct Dyck prefixes). Points form well-separated clusters, with both clusters and induced decision regions colored by the number of unmatched left parentheses. The clear separation across all prefix classes indicates that the representation preserves the full Dyck prefix history, rather than compressing it to low-dimensional summary statistics.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state the three contributions that the paper delivers: (i) a probabilistic framework that mathematically defines task vectors and identifies two inference modes (developed in Sec. 4); (ii) a Bayesian task-retrieval mode whose simplex-projected coefficients track the true posterior (Sec. 5); and (iii) an extrapolative task-learning mode operating in a near-orthogonal subspace that emerges with sufficient task diversity (Sec. 6, Sec. 6.3). All claims are supported by either theorems with proofs (Theorems 1 and 2) or controlled synthetic experiments and a qualitative pretrained-LLM sanity check.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated "Limitations and Future Work" section appears at the end of the main text. It explicitly notes (a) the restrictive first-order Markov assumption underlying `E1`–`E3`, (b) that the orthogonal-subspace finding is descriptive rather than a full mechanistic account of OOD generalization, and (c) that the framework is theory-inspired but not yet a formal theory of generalization or learning dynamics.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results are numbered and cross-referenced. Properties P0–P3 are stated in Sec. 4 with their assumptions; the chain of implications among them is proved in Sec. B.1. Theorem 1 (Bayesian realizability) is stated in Sec. 4.2 with its full hypotheses and proof provided in Sec. B.2. Theorem 2 (simplex packing obstruction) is proved in Sec. B.4. Closed-form expressions underlying mode M2 are derived in Sec. B.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All synthetic data-generation processes are fully specified: E1–E3 are described by closed-form latent priors and conditional distributions in Sec. 3, while the planted-Dyck construction E4 is described in Sec. 7 and expanded in Sec. F. Architecture, optimizer, learning-rate schedule, context length, batch size, and training horizon for every experiment are listed in Tables 3 and 4 in Sec. A. Task-vector extraction, the simplex projection, the orthogonal-subspace optimization, and the causal interventions are described in Sec. C and Sec. D. The pretrained-LLM experiments specify the model (Qwen2.5-7B), prompt format, hidden-state extraction position, task list, OOD task construction, and seeds (Sec. D.5).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/ezyhdxm/mini-ICL`; it contains scripts to generate the synthetic data, train all reported models, extract task vectors, and reproduce the simplex- and orthogonal-subspace analyses and the pretrained-LLM experiments, together with environment specifications. All datasets used in the paper are synthetic and produced by the data-generation procedures described in Sec. 3, except for the pretrained-LLM experiments which use the publicly released Qwen2.5-7B and the function-vectors task collection of Todd et al. [51].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training pipeline (decoder-only transformer with RoPE, AdamW optimizer, autoregressive next-token loss) is summarized in Sec. 3, and all per-experiment

hyperparameters—depth, width, FF dimension, number of heads, context length, batch size, number of optimizer steps, learning rate, weight decay, warmup, schedule, gradient clipping, and minimum learning rate—are tabulated in Tables 3 and 4 in Sec. A. Evaluation protocols (data splits between training latents, minor latents and OOD latents; the prompt format and shot counts for the pretrained-LLM experiments) are described in Sec. D.1, Sec. D.2, and Sec. D.5.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Quantitative figures display variability across evaluation prompts. The simplex-projection and posterior-alignment plots (e.g., Fig. 3 and Figures in Sec. C.4.3) report 10–90$^{\text{th}}$ percentile bands across evaluation prompts for both the simplex coefficients $\beta_{t,k}$ and the ground-truth posteriors $\alpha_{t,k}$. The causal-intervention bar charts in Sec. D.8 use inter-quartile ranges across evaluation batches and compare against random same-rank baselines. The defined source of variability in each case is the random sampling of evaluation prompts (latent and sequence draws) at a fixed model checkpoint.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Sec. A reports that all training and downstream analyses are run on a single NVIDIA RTX 3090 (24 GB) in fp32. Per-experiment runtimes for one training run are given (about 10 min for `E1`, 13 min for `E3` and `E4`, 50 min for `E2`), as is the cost of the task-diversity sweep used in the two-mode analyses (roughly 2–9 GPU-hours per experiment for a single

seed). Pretrained-LLM experiments load each model in `bfloat16` on the same GPU; the per-model analyses of Sec. D.5–D.6 complete in well under one GPU-hour per model, and the persona belief-dynamics study of Sec. D.7 (5,700 prompts per model and format) took under two GPU-hours in total. The finite-order, capacity, and estimator sweeps of Sec. E, Sec. D.3, and Sec. C.6 reuse the training recipe above (about 13 min per `E3`-type run at 30k steps, roughly $3\times$ that at 100k steps).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. The work involves only synthetic data generated from explicit probabilistic models, publicly released pretrained language models used for analysis, and a public persona-statement dataset [36], with no human subjects, no personally identifiable information, and no deployment of new high-risk artifacts.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper is foundational research on the representation geometry of small transformers trained on synthetic data, complemented by observational analysis of existing pretrained models. It does not introduce new capabilities, deployable systems, or datasets and we do not foresee a direct path to societal impact beyond improved scientific understanding of how transformers represent latent tasks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out

that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no pretrained generative models, no scraped datasets, and no high-risk artifacts. The only released materials are research code and synthetic data-generation scripts; the small transformers trained in this study (6–16 layers, $d = 128$) are far below any meaningful misuse threshold.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All third-party assets are cited in the references: the Qwen2.5 and Qwen3 models [38] (Apache 2.0), Llama 3 and Llama 3.1 [12] (Llama community license), Gemma 3, Mistral and Ministral, OLMo 3, SmolLM3, and Phi-4-mini (each under its published license, used for inference-only analysis; model identifiers are listed in Sec. D.6), the HuggingFace `transformers` library [58] (Apache 2.0), the function-vectors task collection of Todd et al. [51] (MIT), and the persona datasets of the model-written evaluations of Perez et al. [36] (CC-BY-4.0). Each is used within the terms of its license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The supplementary code release contains documented scripts for synthetic data generation, model training, task-vector extraction, and the simplex/orthogonal-subspace analyses, together with a README describing dependencies, entry points, and how to reproduce each main-text figure. No new datasets or pretrained model weights with potential consent issues are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or any research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or any research with human subjects, so no IRB review was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology of the paper—the latent-variable framework, the task-vector properties P0–P3, and all theoretical results—is developed independently of any LLM. Pretrained Qwen2.5-7B is used in a standard probing setup as an empirical validation step, not as a component of the methodology, and any incidental use of LLMs for writing or editing did not affect the scientific content of the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.