THE SHARED GEOMETRY OF LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models with different sizes and architectures develop strikingly similar representation geometry, with specific observed arrangements such as cycles and helices often interpreted as traces of computation. We propose the simpler explanation that much of this shared geometry reflects the geometry at model input and output. We show that the shape of input embeddings extends into middle layers while representations converge to the geometry of their predictions well before the last layer. To understand what drives the geometry of these input and output weights, we train models without hidden layers or with only one attention layer and show that they recover a large fraction of the geometry shared across models. This suggests that low-order corpus statistics are a major source of the shared input and output geometry. We support this claim with tokenizer interventions that produce predictable geometric changes in both shallow and deeper models. Finally, we trace six previously reported structures in LLMs to the shape of the input or the shape of the answer. This suggests input and output geometry as an alternative explanation to rule out before attributing representation geometry to task-specific computation.

1 INTRODUCTION

Large language models (LLMs) that differ in architecture and capability exhibit surprisingly similar geometry in their hidden representations (Lee et al., 2025; Hu et al., 2026; Jalouzot et al., 2026; Prieto et al., 2026). Many semantically meaningful geometries and manifolds have been documented in LLM representations, yet we lack a general account of their origin and of whether their presence implies a role in task-specific computation (Engels et al., 2025; Kantamneni & Tegmark, 2025).

Shared geometry is best established at the endpoints of the network, where different LLM families learn aligned embedding and unembedding matrices (Lee et al., 2025). We find that a transformer with only a *single attention layer* shares this geometry too, reproducing specific structures previously studied in LLMs, such as circular arrangements of months and the Fourier patterns of integers observed in arithmetic tasks (Engels et al., 2025; Kantamneni & Tegmark, 2025).

One-layer attention-only transformers predict through direct bigram and attention-mediated skip-trigram paths (Elhage et al., 2021), so their recovery of this geometry points to low-order corpus statistics as a source. We validate this with simple tokenizer interventions that alter token bigram statistics while leaving the corpus unchanged, showing they produce predictable changes in the geometry of both shallow and deeper models. This extends the connection between word embeddings and corpus co-occurrence (Levy & Goldberg, 2014; Pennington et al., 2014; Karkada et al., 2026; Prieto et al., 2026) to the input and output interfaces of autoregressive LLMs.

Similar embedding and unembedding weights mean that first-layer representations share geometry when encoding the same tokens, and final-layer representations share geometry when predicting the same tokens. We show that this shared endpoint geometry strongly shapes hidden representations throughout the network. We compare hidden states with two reference geometries, the *shape of the input*, given by current-token embeddings, and the *shape of the answer*, given by the model's predictions pulled back through the unembedding. The shape of the input persists through early and middle layers, and the shape of the answer emerges well before the final layer. Together, these two components form a simple account of the origin of representation geometry and help explain why it is shared across model families.

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

Our account traces disparate observations of representation geometry to a common origin, identifying previously unexplained structure in prior work as either the shape of the input or the shape of the answer. It thus provides an alternative explanation to test before attributing observed structure to task-specific computation. **Our contributions are:**

- We show that transformers with only one attention layer recover much of the shared geometry across LLMs at the embedding and unembedding.
- We demonstrate that tokenizer interventions that alter token bigram statistics produce predictable changes in the endpoint geometry, connecting this shared structure to low-order corpus statistics.
- We show that the agreement between LLMs at the input and output extends to the hidden representations. Early layers remain similar to the embedding while late layers become increasingly similar to the eventual predictions, providing an explanation for shared geometry across LLMs.
- We show that previously studied structures in LLM representations can be explained as the shape of the input or the shape of the answer.

We will make our implementation openly available upon publication.

2 Setup

Language models. This work studies representation geometry in language models. Specifically, we study a set of open-weight models, $\mathcal{M} = \{$Llama-3.1-8B, Qwen2.5-7B, Pythia-2.8B, OLMo-2-7B$\}$ (Grattafiori et al., 2024; Qwen et al., 2024; Biderman et al., 2023; Team OLMo et al., 2025). In addition, as controlled experiments we train transformers with 0, 1 and 16 layers on a dataset of 8B tokens from the Pile (Gao et al., 2021). Our trained models use the GPT-2 tokenizer, pre-norm RMSNorm, and SwiGLU (Shazeer, 2020). We denote model m's embedding and unembedding matrices by $\boldsymbol{E}_m$ and $\boldsymbol{U}_m$, respectively.

Consensus geometry across models. We measure the geometry of d-dimensional model representations or embedding/unembedding vectors $\mathbf{Z}_m \in \mathbb{R}^{n\times d}$ using the dot-product kernel $\mathbf{K}_m := K(\boldsymbol{Z}_m) = \mathrm{unit}(\boldsymbol{Z}_m\boldsymbol{Z}_m^\top)$, which identifies $\mathbf{Z}_m$ up to orthogonal transformation and global scale (Dokmanic et al., 2015), with $\mathrm{unit}(\boldsymbol{A}) = \boldsymbol{A}/\|\boldsymbol{A}\|_F$. For a set of model families $\mathcal{M}$, with one representative model per family, and the collection $\boldsymbol{Z} = (\boldsymbol{Z}_m)_{m\in\mathcal{M}}$, we define the *consensus geometry*

$$K_{\mathcal{M}}(\boldsymbol{Z}) = \mathrm{unit}\Big(\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} K(\boldsymbol{Z}_m)\Big), \tag{1}$$

Positive rescaling and averaging preserve positive semidefiniteness, so this *consensus geometry* is itself a kernel. When kernels are computed on embedding or unembedding vectors, we use the notation $K_{\mathcal{M}}(\boldsymbol{E})$ and $K_{\mathcal{M}}(\boldsymbol{U})$, respectively, where $\boldsymbol{E} = (\boldsymbol{E}_m)_{m\in\mathcal{M}}$ and $\boldsymbol{U} = (\boldsymbol{U}_m)_{m\in\mathcal{M}}$.

Comparing representation geometry. We measure agreement between kernels $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ in two ways. Global agreement is measured with linear centered kernel alignment (CKA; Kornblith et al., 2019), computed with the unbiased HSIC estimator (Song et al., 2012). Writing $\tilde{\boldsymbol{K}}_1$ and $\tilde{\boldsymbol{K}}_2$ for the U-centered kernels (Appendix A.1),

$$\mathrm{CKA}(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{\langle \tilde{\boldsymbol{K}}_1,\ \tilde{\boldsymbol{K}}_2\rangle_F}{\|\tilde{\boldsymbol{K}}_1\|_F\,\|\tilde{\boldsymbol{K}}_2\|_F} \tag{2}$$

the cosine similarity between centered kernels. U-centering removes the diagonal and row effects that bias the standard estimator at finite n.

Because CKA is dominated by high-variance directions and can be insensitive to lower-variance structure (Kornblith et al., 2019; Ding et al., 2021), we also measure agreement in local geometry using the chance-corrected fraction of shared nearest neighbors (Huh et al., 2024).. Let $\mathcal{N}_k^{\boldsymbol{K}}(i)$ be the k points $j \neq i$ closest to i under kernel $\boldsymbol{K}$, i.e. with the smallest squared distance $K_{ii} + K_{jj} - 2K_{ij}$. With $o_k(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{1}{nk}\sum_{i=1}^n \big|\mathcal{N}_k^{\boldsymbol{K}_1}(i) \cap \mathcal{N}_k^{\boldsymbol{K}_2}(i)\big|$ the mean neighborhood overlap,

$$\mathrm{N@}k(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{o_k(\boldsymbol{K}_1, \boldsymbol{K}_2) - \frac{k}{n-1}}{1 - \frac{k}{n-1}}, \tag{3}$$

where $k/(n-1)$ is the expected overlap of random neighbor sets, so that 0 corresponds to chance and 1 to identical neighborhoods (Appendices A.2 and B.2).

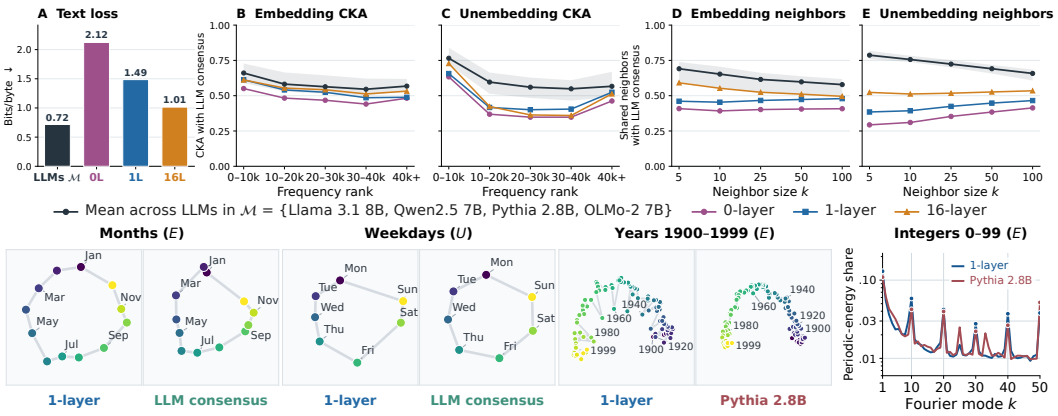


Figure 1: **Shallow models recover embedding and unembedding geometry shared across LLMs.** Despite a large difference in predictive performance (**A**) models with one or no attention layers recover most of the geometry shared across LLMs, as measured by CKA **(B-C)** and chance corrected shared nearest neighbors **(D-E)**. The 1-layer models also captures the same structures as the LLM consensus for groups of tokens like months days, years or integers. We compare to Pythia for integer due to different tokenization in other models.

3 Shared geometry at the input and output weights

In order to understand the origins of representation geometry in LLMs we begin by studying the embedding and unembedding weights. These weights provide an interface between discrete tokens and hidden representations and exhibit similar geometry across model families (Lee et al., 2025). We characterize this shared interface geometry and ask how much of it is recovered by shallow models.

Consensus geometry at the input and output Using the methodology outlined in Section 2, we define the consensus geometry for a collection of embeddings $\boldsymbol{E}$ and unembedding $\boldsymbol{U}^\top$ across a set of open source LLMs $\mathcal{M}$. The resulting average kernels $K_{\mathcal{M}}(\boldsymbol{E})$ and $K_{\mathcal{M}}(\boldsymbol{U}^\top)$ capture the structure shared across these LLMs at the input and output weights. We find that each reference model agrees strongly with the consensus geometry of the other families (Figure 1B–E), validating the results in Lee et al. (2025). This agreement holds across frequency bands (Figure 1B–C) and extends to local neighborhoods (Figure 1D–E) and specific arrangements previously observed in language models (Figure 1(bottom row)).

Shallow models recover shared interface geometry To test whether this shared geometry appears only in large language models, we train a model with one attention layer and a model with no hidden layers at all on 8B Pile tokens (Gao et al., 2021) to measure how much of the geometry shared by LLMs they capture (details in Appendix B.3). We also train a 16-layer model on this same data to see how much additional geometry appears with scale when training on the same data.

Our results show that the single-layer model agrees with the cross-LLM consensus geometry *almost as much as the average LLM* at both $\boldsymbol{E}$ and $\boldsymbol{U}$ (Figure 1B–E), despite a large gap in predictive performance (Figure 1A). This recovery extends to local dot-product neighborhoods, with output-side overlap approaching the reference-family mean. Interestingly, the structure captured at the input and output weights of 1-layer models extends to specific arrangements that have previously been observed in the hidden representations of LLMs. For example, Figure 1(bottom) shows cyclic arrangements for months and days, ordered year representations and a specific Fourier structure very similar to those observed in the LLM consensus at $\boldsymbol{E}$ and $\boldsymbol{U}$ but also in hidden representations of LLMs in previous work (Engels et al., 2025; Kantamneni & Tegmark, 2025).

While we have demonstrated that (1) these patterns appear in the embedding and unembedding weights and that (2) they are shared by 1-layer models, previous literature usually studies these patterns in the intermediate layers of neural networks. However, if it is the case that the intermediate patterns are caused by these endpoint geometries captured even by 1-layer models, this would mean complex computation is not required to create these patterns, narrowing the possible origins to bigram and skip-trigram statistics of the data (the origins of this geometry are explored in more detail in

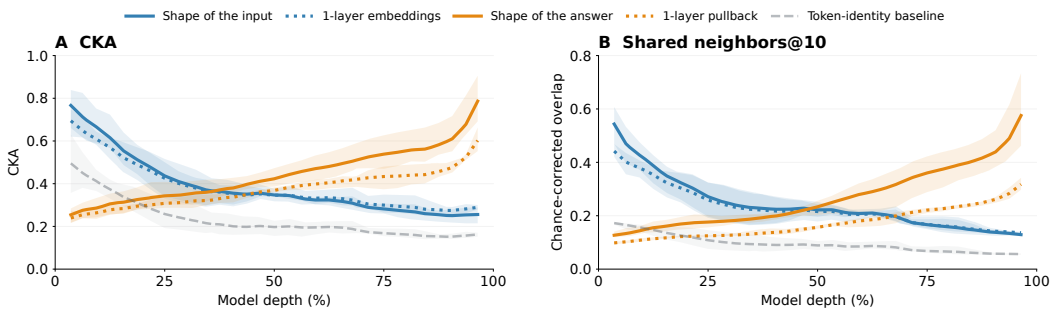


Figure 2: **Input and answer geometry extend beyond the endpoints.** Agreement with input geometry persists beyond the first blocks, while agreement with answer geometry appears before the final readout. Lines average direct reference–state agreement across the four models in $\mathcal{M}$ on 2,041 shared FineWeb contexts and bands show the range across models. Dotted curves replace each model's own endpoints with endpoints derived from a 1-layer model and independent predictions.

Section 6). For this reason we now study how geometry at the input and the output can impact geometry in hidden representations.

4 Input and Output Geometry in Hidden Representations

Layers in LLMs are connected by residual connections, which create an identity path from the first layer to the last. Any structure introduced by the embeddings therefore remains in the residual stream unless later layers overwrite it. Likewise, structure in late layers is carried forward into the final prediction unless it is overwritten. Indeed, applying the unembedding to intermediate layers already yields reasonable predictions (nostalgebraist, 2020; Belrose et al., 2023), suggesting that LLMs represent the answer well before the last layer. We study the geometric counterpart of these observations by measuring how far into the network the embeddings shape representations, and at what point representations start reflecting the shape of the eventual predictions. Since embeddings and unembeddings are shared across model families (Section 3), the answer also tells us how much of the geometry of hidden representations is shared as a result.

4.1 The shape of the input and the shape of the answer

For contexts $\mathcal{C}$ ending in tokens $\mathbf{x}_\mathcal{C} = (x_1, \ldots, x_n)$, let $\boldsymbol{H}_{m,\ell}(\mathcal{C})$ stack model m's normalized final-position hidden states after layer ℓ, with N_m total layers. We use 2,041 matched natural-text prefixes from FineWeb (Penedo et al., 2024) across the four models in $\mathcal{M}$ (Appendix C.1). We characterize the geometry $K(\boldsymbol{H}_{m,\ell})$ at each layer by its agreement with the geometry at the first and last layer: the *shape of the input* $K(\boldsymbol{H}_{m,0})$ and the *shape of the answer* $K(\boldsymbol{H}_{m,N_m})$. Our results in Figure 2 show that early layer representations retain similarity with the shape of the input while late layer representations become increasingly similar to the shape of the answer.

While these results demonstrate that geometry at the endpoints strongly influence that in hidden representations, we can go further to show the influence of embedding/unembedding weight structure that was shown above to be shared by even the simple 1-hidden layer models. For this we note that the shape of the input and the shape of the answer can be written in terms of the embeddings $\boldsymbol{E}$ and unembeddings $\boldsymbol{U}$ whose geometry we studied in Section 3:

$$\boldsymbol{H}_{m,0}(\mathcal{C}) = \rho_m\big(\boldsymbol{E}_m[\mathbf{x}_\mathcal{C}]\big), \qquad \boldsymbol{H}_{m,N_m}(\mathcal{C}) = \bar{\boldsymbol{L}}_m(\mathcal{C})\,\overline{\boldsymbol{U}}_m^{+\top}, \tag{4}$$

where ρ_m is the RMS normalization, $\bar{\boldsymbol{L}}_m(\mathcal{C})$ are the model's centered logits with any fixed readout offset removed, and $\overline{\boldsymbol{U}}_m$ is its centered effective unembedding (Appendix C.1). The first identity holds because the models add no positional vector to the embedding. The second holds because the normalized final state produces the logits linearly through a readout of full column rank.

Having measured the similarity of each model's representations with it's own first and last layer, we now see how much of this agreement survives if we use Equation (4) to replace each endpoint

with one supplied by a different model. On the input side, we use the normalized embeddings of the one-layer model from Section 3, $K(\rho_{1L}(\boldsymbol{E}_{1L}[\mathbf{x}_{\mathcal{C}}]))$. Their agreement with early layers is close to that of the models' own embeddings: much of the input structure preserved inside these networks is already learned by a shallow model. On the output side, we pull back the logits of an independent model, Mistral-Nemo Base 12B (Mistral AI, 2024), through the one-layer model's unembedding, $K(\bar{\boldsymbol{L}}_{\text{Nemo}}(\mathcal{C})\,\overline{\boldsymbol{U}}_{1L}^{+\top})$.

These results show that the same geometry that is captured by even 1-layer models is present in early and late hidden representations, suggesting an explanation for why the same structures observed at E and U in Figure 1 appear in hidden representations of LLMs (Engels et al., 2025). To verify this, we now analyze previously studied structures in LLMs and ask if they can be explained in terms of the shape of the input and the shape of the answer.

5 A COMMON ORIGIN OF PREVIOUSLY OBSERVED STRUCTURE

Several structures observed in LLM hidden representations have been interpreted as supporting specific computations, such as cycles of months and weekdays for modular arithmetic and periodic integer structure for addition (Engels et al., 2025; Kantamneni & Tegmark, 2025). However, showing that a set of representations is used in a computation does not imply that the computation explains their specific arrangement. A circuit can use an arrangement that was already in place, for example one inherited from embeddings whose geometry reflects low-order corpus statistics (Section 6). We offer a concrete alternative explanation for where such arrangements come from. They can already be present in the current-token embedding, or arise from the geometry of the model's predictions pulled back through the unembedding. For example, a one-layer attention-only model, which cannot compose the steps that calendar arithmetic requires, already arranges month, weekday, and year embeddings as they appear inside LLMs (Figure 1(bottom row)).

Section 4 showed that embedding geometry persists into early and middle layers and that prediction geometry appears well before the final layer. This lets us test whether a particular structure in hidden representations can be explained by the shape of the input or the shape of the answer. We say that a structure observed at layer ℓ is *explained by the shape of the input* if the same structure is also present in every previous layer and the embedding weights. Symmetrically, it is *explained by the shape of the answer* if the same structure appears at every layer from ℓ to the final layer. This test provides a stronger null hypothesis to rule out before attributing functional origins to specific representations. We apply this test to six previously reported structures and report our findings inTable 1. In the following pages, we study two of these findings in further detail, and present the detailed accounts for the rest in Appendix D.

Table 1: **Previously reported structures are explained by the shape of the input, the shape of the answer, or both.** Each row names the structure, its origin per our test, and the section supporting it.

Reference	Tested structure	Origin	Details
Feucht et al. (2026)	Calendar arithmetic transition	Input and answer	Section 5.1
Wurgaft et al. (2026)	Calendar answer manifolds	Shape of the answer	Section 5.1
Shai et al. (2024)	Belief-state geometry	Shape of the answer	Section 5.2
Engels et al. (2025)	Calendar circles	Shape of the input	Appendix D.1
Kantamneni & Tegmark (2025)	Periodic number structure	Shape of the input	Appendix D.2
Park et al. (2025a)	Persistent weekday order	Shape of the input	Appendix D.6

5.1 CALENDAR ARITHMETIC

Calendar related concepts like months and days have been studied in detail due to their ordered and cyclic representations across LLMs. This was first found by Engels et al. (2025), who suggested this arrangement may be due to models performing arithmetic with months or days. However Feucht et al. (2026) note that this cyclic structure appears after the key layers involved in arithmetic computation suggesting this structure needs an alternative explanation. We therefore study the same setup used in Feucht et al. (2026) and show that cyclic structures in late layers like the ones found in Feucht et al. (2026); Wurgaft et al. (2026); Engels et al. (2025) can be explained as the *shape of the answer*.

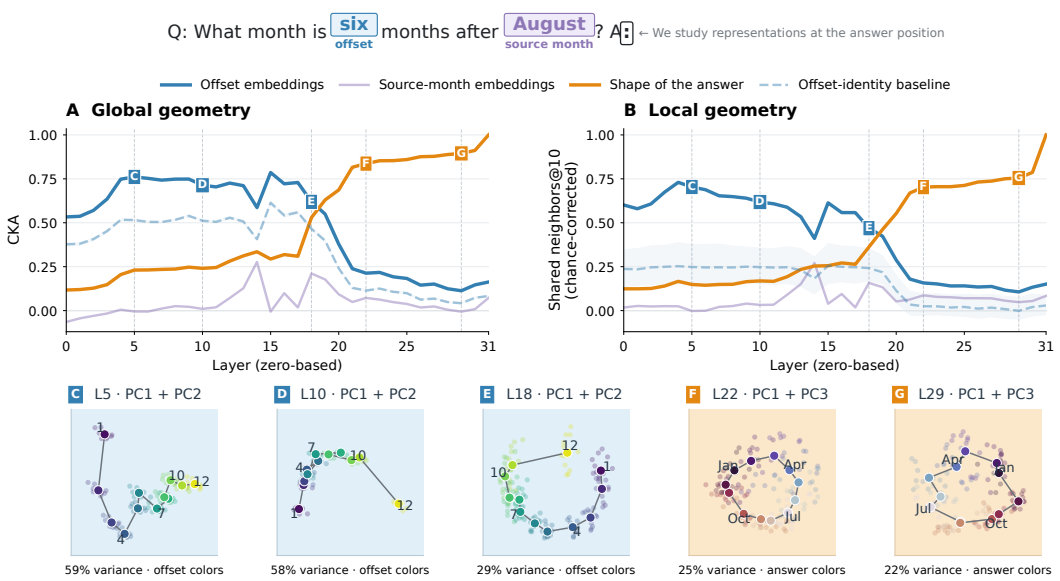


Figure 3: **Input and output geometry in month arithmetic. A–B:** CKA and neighbors@10 both show that early layers reflect the embedding geometry of the offset involved in the prompt while late layers reflect the geometry of the predictions each prompt makes through the unembedding interface. PCA plots confirm that prompts are grouped by the offset they contain in early layers (**C–E**) and grouped by the month they predict in late layers (**F–G**).

We study the activations of Llama 3.1 8B when performing month arithmetic with prompts like 'Q: What is six months after August? A:' as in Feucht et al. (2026); Wurgaft et al. (2026). Similarly to Section 4, we compare its geometry with fixed reference kernels, but the input references now come from earlier tokens in the sequence:

$$\boldsymbol{K}_{\text{offset}} = K\big(\rho(\boldsymbol{E}[\mathbf{x}_o])\big), \qquad \boldsymbol{K}_{\text{month}} = K\big(\rho(\boldsymbol{E}[\mathbf{x}_s])\big), \qquad \boldsymbol{K}_{\text{answer}} = K\big(\bar{\boldsymbol{L}}\,\overline{\boldsymbol{U}}^{+\top}\big), \tag{5}$$

where $\mathbf{x}_o$ and $\mathbf{x}_s$ stack the offset and source-month tokens of the 144 prompts and $\bar{\boldsymbol{L}}$ their centered final logits.

Both global CKA and local-neighborhood agreement show the same progression (Figure 3A–B). Offset-embedding geometry dominates through early and middle layers, while answer geometry overtakes it around layer 19 and becomes increasingly prominent thereafter. This transition is strong enough that it is even visible in PCA plots, which show representations reflect offset embedding geometry at early layers and the shape of the final predictions after layer 18.

Attributing late layer cyclic structure in month representations to the shape of the answer explains why this structure appears after the key layers involved in arithmetic (Feucht et al., 2026) and why Wurgaft et al. (2026) find an isometry between the representations in layer 28 and model predictions. The latter is to be expected if these late layer representations are preserved through all later layers and mapped to logits by a linear unembedding. We also study in more detail how specific structures in month, years, and day representation studied at layer 7 of GPT2 by Engels et al. (2025) can similarly be explained by the shape of the input in Appendix D.1.

5.2 Two sources of answer geometry

While the shape of the input depends only on embeddings, the shape of the answer depends on two factors, the geometry of the target logits and the unembeddings through which these predictions are made. In Figure 4 we show two tasks which provide examples of geometry required by the target logits and geometry imposed by the unembedding interface, while also showing that these specific arrangements in late layer representations can be explained as the shape of the answer.

We start with the Mess3 task used in Shai et al. (2024) as an example that optimal logits for a task can have a rich structure which appears in the late layers of models trained on this task. This specific

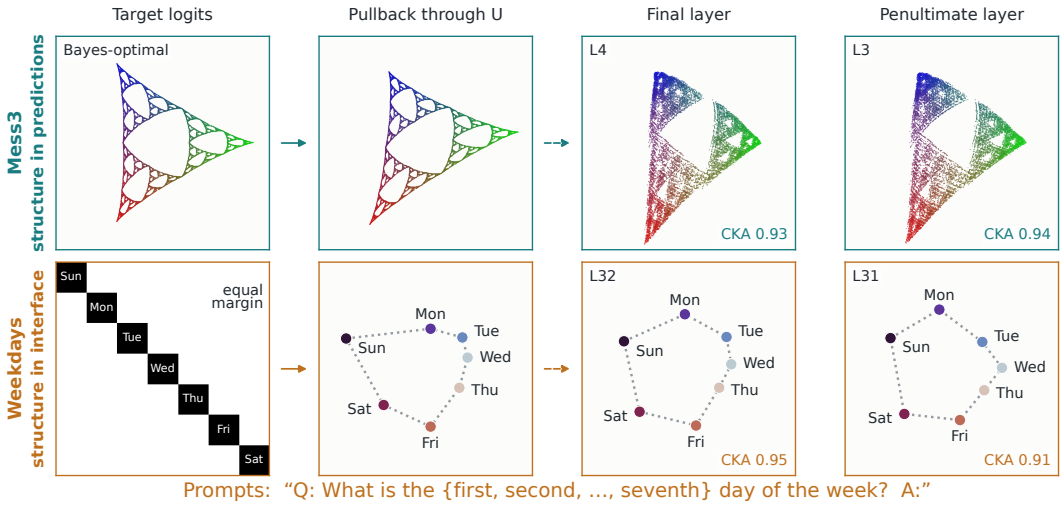


Figure 4: **A:** Mess3 Bayes-optimal logits already have structured geometry, their pullback preserves it and agrees with final and penultimate states. **B:** One-hot weekday logit targets have no preferred cyclic ordering yet their pullback through $\boldsymbol{U}$ acquires a cyclic component that aligns with late states during weekday retrieval.

task is designed so that the optimal predictions approximate a fractal structure and Figure 4 shows that the same structure is present in the last two layers of a model trained on this task (CKA 0.93 and 0.94), providing an example of the shape of the answer appearing before the last layer.

Weekday retrieval (Figure 4B) isolates the unembedding weights as a source of structure. We prompt Llama-3.1-8B with 'Q: What is the first day of the week? A:', varying the ordinal from first to seventh. As target logits we use one-hot vectors for the correct day, so every pair of targets is equally similar, and the targets themselves encode no ordering of the days. Any cyclic structure in their pullback must therefore come from the unembedding. The pullback does form a cycle, and it matches Llama's final and penultimate layers (CKA 0.95 and 0.91).

Attributing these two examples of representation geometry to the shape of the answer explains why the structure studied in Shai et al. (2024) is linearly represented, since it needs to be mapped to the logits by a linear map. Additionally, it shows that cyclic arrangements of days in late layers can result from alignment with the cyclic structure already present in unembedding weights and that this appears for tasks that involve predicting days even when they do not involve any arithmetic.

6 Low-Order Prediction Shapes the Shared Interfaces

The preceding sections show how shared input and output geometry helps explain structure in hidden representations and recurring geometric patterns. We now investigate what gives these interfaces their particular arrangement. The recovery achieved by a single attention layer (Section 3), and even a zero-layer model, points to low-order prediction as a source. The zero-layer model captures direct bigram relationships, while one attention layer introduces attention-mediated skip-trigram paths. We use this structure to motivate two interventions on local prediction statistics and test whether the resulting geometric changes extend to deeper models.

Decomposing one attention layer into bigrams and skip trigrams. A single attention layer predicts through a direct token-to-token path and attention-mediated paths linking a source token, the current token, and a possible continuation (Elhage et al., 2021). Let $\boldsymbol{e}(x)$ be the embedding of token x and $\boldsymbol{U}$ the output matrix. Suppressing normalization and positional terms, the logits $\boldsymbol{z}_t$ at position t take the form

$$\boldsymbol{z}_t = \underbrace{\boldsymbol{U}\boldsymbol{e}(x_t)}_{\text{direct bigram path}} + \underbrace{\sum_h \sum_{j \le t} \alpha_{tj}^{(h)} \boldsymbol{U}\boldsymbol{W}_{\mathrm{OV}}^{(h)} \boldsymbol{e}(x_j)}_{\text{attention-mediated skip-trigram paths}} . \tag{6}$$

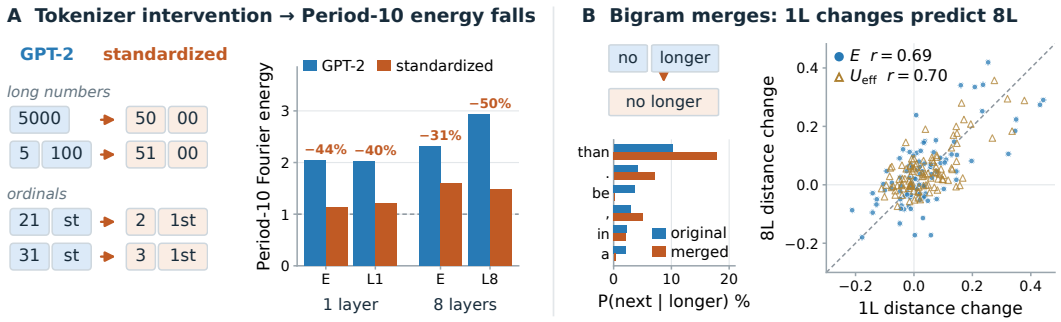


Figure 5: **Two interventions connect local prediction statistics to geometry. A** Standardizing the GPT2 tokenizer so only one and two-digit numbers have their own token and adding 13 ordinal tokens leads to a sharp decrease (31-50%) in period 10 energy in the representations of integers across layers. This effects holds for both a 1 and an 8-layer transformer trained on data tokenized with this intervention. **B** Beyond integer embeddings we introduce 200 new bigrams that alter the continuations of existing tokens. We find a correlation of $r = 0.69$ at $\boldsymbol{E}$ and $r = 0.70$ at effective $\boldsymbol{U}$ between the changes in the geometry of a 1 and 8-layer model.

Here h indexes heads, $\alpha_{tj}^{(h)}$ is the attention weight from position t to position j, and $\boldsymbol{W}_{\mathrm{OV}}^{(h)}$ is the head's value-to-output map. The direct path supplies a low-rank bigram contribution. Each attention path selects a source token and uses it to adjust the next-token logits.

Given our results in Section 3, this connection of shallow models to corpus level statistics suggests the hypothesis that geometry shared across models like the Fourier structure in integer representations is a byproduct of compressing bigram and skip-trigram statistics rather than more complex computation like arithmetic (Kantamneni & Tegmark, 2025).

Changing local prediction statistics changes token geometry. We test this hypothesis with two interventions which change the tokenizer to alter bigram statistics while leaving the corpus unchanged. The first intervention is aimed at the Fourier structure in integer embeddings that Kantamneni & Tegmark (2025) attribute arithmetic. We target the period-10 Fourier energy in representations of integers with a leading space. The standard GPT-2 tokenizer stores some four-digit numbers as single tokens but splits others irregularly. Our intervention simply assigns a token to all 1 and 2-digit numbers and none to larger numbers. We also make a change to the tokenization of ordinal suffixes ([`2`, `2nd`] instead of `22nd`).

We train two models with 1 and 8 layers on 2B tokens of the Pile with a regular GPT2 tokenizer and the same data with our tokenizer intervention. Both models lose between 31 and 50% of their period 10 Fourier energy, in line with a drop of 35% in corpus bigram statistics (Figure 5A).

As a broader intervention beyond integer embeddings, we introduce 200 tokens for bigrams selected to maximally alter the bigram statistics of remaining tokens. For example, making `no longer` a single token changes the remaining uses of `longer`: `be` becomes less likely and `than` more likely as its next token. We train one- and eight-layer models on the same 2B-token Pile prefix with the original and expanded tokenizers. On 100 preselected high-effect token pairs at each interface, the one-layer distance changes predict the direction and magnitude of the eight-layer changes (Figure 5D–E). Pair selection is frozen before the eight-layer run (Appendix E.2).

7 Related Work

Shared representation geometry. Independently trained networks share relational structure (Lu et al., 2018; Kornblith et al., 2019), enabling cross-model transfer (Moschella et al., 2023; Jha et al., 2025). The Platonic representation hypothesis interprets convergence as learning a shared statistical model of reality (Huh et al., 2024). Language-model studies document agreement in embedding and unembedding geometry (Lee et al., 2025), contextual concept manifolds (Hu et al., 2026), and linguistic organization across architectures (Jalouzot et al., 2026). Examples include spatial, hierarchical, and calendar structures (Gurnee & Tegmark, 2024; Park et al., 2025b; Engels

et al., 2025). Modell et al. (2025) characterize when feature manifolds preserve topology and intrinsic distances under correspondence and local-similarity assumptions. Such correspondence does not specify which training constraints produce the shared metric. Gröger et al. (2026) further show why global convergence must be calibrated against measurement effects. We ask how much cross-family geometry, measured globally and locally, a shallow predictor can recover.

Statistical and predictive origins. Classical word embeddings connect geometry to corpus co-occurrence factorization (Pennington et al., 2014; Levy & Goldberg, 2014), and recent work studies the resulting symmetries and arrangements (Karkada et al., 2026). Next-token prediction analyses link learned representations to continuation support, semantic concepts, and label structure (Zhao et al., 2024; Zhao & Thrampoulidis, 2025; Jiang et al., 2024; Wu & Papyan, 2024). Prieto et al. (2026) show that correlated bag-of-words data induce semantic clusters and calendar cycles in autoencoders through constructive interference, distinguishing these arrangements from task-driven value codes. Training studies also find that low-order statistics are learned early (Refinetti et al., 2023; Belrose et al., 2024; Chang et al., 2024). For autoregressive transformers, Elhage et al. (2021) decompose one-layer attention-only prediction into bigram and skip-trigram paths and Chang & Bergen (2025) identify small bigram subnetworks within deep models. We connect these low-order mechanisms to cross-family geometric agreement and test whether changes in continuation associations and tokenization produce corresponding geometric changes across model depths.

Prediction geometry and accessible information. Intermediate vocabulary readouts expose provisional predictions (Belrose et al., 2023; Geva et al., 2022; Dar et al., 2023), while controlled sequence models connect prediction and belief-state geometry (Shai et al., 2024; Piotrowski et al., 2025). Our output-pullback account relates to unembedding-based causal metrics (Park et al., 2024) and the readout rank constraints studied in the softmax-bottleneck literature (Yang et al., 2018; Demeter et al., 2020), but asks how a specified prediction is represented through a fixed interface. Geometric prominence need not track computational importance, as sentence similarities exhibit lexical biases (Nikolaev & Padó, 2023) and feature complexity and learning order bias representational variance (Hermann & Lampinen, 2020; Lampinen et al., 2024).

8 Conclusion

We have argued that much of the geometry recurring across language models has a common predictive origin. Bigram and skip-trigram relationships in the training data help shape similar embedding and unembedding weights across models so that even 1-layer models recover most of the geometry shared across LLMs. This geometry recovered even by 1-layer models at the input and output weights includes specific arrangements for months, days, years and integers that had previously been observed in the hidden representations of LLMs. We then show how these specific arrangements present at the input and output weights can also appear in early and late layer representations providing an alternative explanation for previously observed structures in LLMs and a stronger null hypothesis to rule out before attributing the origin of a specific structure to a specific functionality.

Limitations and future work. Our framework explains some of the regularities between language model representations yet does not provide a full account of representation geometry. Future work could provide a formal argument for why even 1-layer models converge to similar embeddings as LLMs with billions of parameters. We also show in Section 5 that the shape of the input can be preserved when being mapped through attention. This suggests that embedding geometry could influence representation geometry through attention, going beyond the direct, same-token path studied in this work.

Identifying the shape of the input and the shape of the answer as important drivers of representation geometry may also help improve interpretability methods. Principal components, probes, and sparse autoencoders can emphasize geometrically prominent structure that is inherited from the input or induced by the answer. Factoring out these expected components could expose lower-variance task-specific structure that might otherwise be dismissed as irrelevant. We leave the development of such methods to future work.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with coding, writing, figure editing and feedback on how to structure and explain parts of the paper. We have not used generative AI tools to generate any of the main experiments or ideas in this work, and synthetic datasets and proofs are not applicable to this work. Additionally, we used generative AI tools for additional literature review. We have reviewed all AI-assisted work. AI assisted literature review only complemented a previous, manual review and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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. URL https://arxiv.org/abs/2303.08112v1.

Nora Belrose, Quintin Pope, Lucia Quirke, Alex Mallen, and Xiaoli Fern. Neural networks learn statistics of increasing complexity. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.04362.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Tyler A. Chang and Benjamin K. Bergen. Bigram subnetworks: Mapping to next tokens in transformer language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.15471.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. Characterizing learning curves during language model pre-training: Learning, forgetting, and stability. *Transactions of the Association for Computational Linguistics*, 12, 2024. URL https://arxiv.org/abs/2308.15419.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pp. 693–703, 2002. doi: 10.1007/3-540-45465-9_59.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 16124–16170, 2023. doi: 10.18653/v1/2023.acl-long.893. URL https://aclanthology.org/2023.acl-long.893/.

David Demeter, Gregory Kimmel, and Doug Downey. Stolen probability: A structural weakness of neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.acl-main.198/.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/0c0bf917c7942b5a08df71f9da626f97-Abstract.html.

Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015. doi: 10.1109/MSP.2015.2398954.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021. doi: 10.1162/tacl_a_00410. URL https://aclanthology.org/2021.tacl-1.60/.

Nelson Elhage et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

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

Sheridan Feucht, Tal Haklay, Usha Bhalla, Daniel Wurgaft, Can Rager, Raphaël Sarfati, Jack Merullo, Thomas McGrath, Owen Lewis, Ekdeep Singh Lubana, Thomas Fel, and Atticus Geiger. Arithmetic in the wild: Llama uses base-10 addition to reason about cyclic concepts. *arXiv preprint arXiv:2605.01148*, 2026. URL https://arxiv.org/abs/2605.01148.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2021. URL https://arxiv.org/abs/2101.00027.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, 2022. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.14486.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.02207.

Katherine L. Hermann and Andrew K. Lampinen. What shapes feature representations? Exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.12433.

Zhimin Hu, Lanhao Niu, and Sashank Varma. Language models represent and transform concepts with shared geometry. *arXiv preprint arXiv:2607.04525*, 2026. URL https://arxiv.org/abs/2607.04525.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Louis Jalouzot, Christophe Pallier, Emmanuel Chemla, and Yair Lakretz. What makes two language models think alike? *arXiv preprint arXiv:2406.12620*, 2026. URL https://arxiv.org/abs/2406.12620v3.

Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the universal geometry of embeddings. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=jiCLUPq5xv.

Yibo Jiang, Goutham Rajendran, Pradeep Kumar Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21879–21911, 2024. URL https://proceedings.mlr.press/v235/jiang24d.html.

Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. *arXiv preprint arXiv:2502.00873*, 2025. URL https://arxiv.org/abs/2502.00873.

Dhruva Karkada, Daniel J. Korchinski, Andres Nava, Matthieu Wyart, and Yasaman Bahri. Symmetry in language statistics shapes the geometry of model representations. *arXiv preprint arXiv:2602.15029*, 2026. URL https://arxiv.org/abs/2602.15029.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Andrew Kyle Lampinen, Stephanie C. Y. Chan, and Katherine Hermann. Learned feature representations are biased by complexity, learning order, position, and more. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2405.05847.

Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=aJDykpJAYF.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27, 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/b78666971ceae55a8e87efb7cbfd9ad4-Abstract.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Qihong Lu, Po-Hsuan Chen, Jonathan W. Pillow, Peter J. Ramadge, Kenneth A. Norman, and Uri Hasson. Shared representational geometry across neural networks. *arXiv preprint arXiv:1811.11684*, 2018. URL https://arxiv.org/abs/1811.11684.

Mistral AI. Mistral NeMo. Model release, 2024. URL https://mistral.ai/news/mistral-nemo/.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*, 2025. URL https://arxiv.org/abs/2505.18235.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.15430.

Dmitry Nikolaev and Sebastian Padó. Representation biases in sentence transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3701–3716, 2023. doi: 10.18653/v1/2023.eacl-main.268. URL https://aclanthology.org/2023.eacl-main.268/.

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

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. ICLR: In-context learning of representations. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2501.00070.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2311.03658.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The geometry of categorical and hierarchical concepts in large language models. In *International Conference on Learning Representations*, 2025b. URL https://arxiv.org/abs/2406.01506.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Mateusz Piotrowski, Paul M. Riechers, Daniel Filan, and Adam S. Shai. Constrained belief updates explain geometric structures in transformer representations. *arXiv preprint arXiv:2502.01954*, 2025. URL https://arxiv.org/abs/2502.01954.

Lucas Prieto, Edward Stevinson, Melih Barsbey, Tolga Birdal, and Pedro A. M. Mediano. From data statistics to feature geometry: How correlations shape superposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7akSRQS5Xh.

Qwen, An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. Neural networks trained with SGD learn distributions of increasing complexity. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.11567.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8936fa1691764912d9519e1b5673ea66-Abstract-Conference.html.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(47):1393–1434, 2012. URL https://www.jmlr.org/papers/v13/song12a.html.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, et al. 2 OLMo 2 Furious. *arXiv preprint arXiv:2501.00656*, 2025. URL https://arxiv.org/abs/2501.00656.

Robert Wu and Vardan Papyan. Linguistic collapse: Neural collapse in (large) language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2405.17767.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphaël Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah D. Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior. *arXiv preprint arXiv:2605.05115*, 2026. URL https://arxiv.org/abs/2605.05115.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkwZSG-CZ.

Yize Zhao and Christos Thrampoulidis. Geometry of semantics in next-token prediction: How optimization implicitly organizes linguistic representations. *arXiv preprint arXiv:2505.08348*, 2025. URL https://arxiv.org/abs/2505.08348.

Yize Zhao, Tina Behnia, Vala Vakilian, and Christos Thrampoulidis. Implicit geometry of next-token prediction: From language sparsity patterns to model representations. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=qyilOnIRHI.

APPENDIX

A MEASURING REPRESENTATION GEOMETRY

The main figures compare relationships between matched tokens or contexts, rather than coordinates in different models. This section specifies the two agreement measures and how predictions become a representation-space reference. The following sections give the settings and checks for each main result.

A.1 GLOBAL AGREEMENT: CKA WITH THE HSIC CORRECTION

For representations collected as rows of $\boldsymbol{X}$, the ordinary Gram matrix is $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^\top$. CKA compares the pattern of similarities across all pairs. We use the unbiased HSIC estimator (Song et al., 2012) in the normalized alignment of Kornblith et al. (2019). To specify the correction, set the diagonal of $\boldsymbol{G}$ to zero, obtaining $\boldsymbol{G}^\circ$, and write $r_i = \sum_j G^\circ_{ij}$ and $r_\bullet = \sum_i r_i$. For $n > 3$,

$$[\mathcal{U}_n(\boldsymbol{G})]_{ij} = \begin{cases} G^\circ_{ij} - \dfrac{r_i + r_j}{n-2} + \dfrac{r_\bullet}{(n-1)(n-2)}, & i \neq j, \\ 0, & i = j. \end{cases} \tag{7}$$

We insert these corrected matrices into Equation (2). This removes the diagonal self-similarities and applies the finite-sample HSIC correction. "Unbiased CKA" denotes this normalized HSIC estimator; the normalized ratio itself is not an unbiased estimator. The correction is used only when scoring CKA, not when forming the consensus, plotting PCA, or finding neighbors.

A.2 LOCAL AGREEMENT: SHARED NEAREST NEIGHBORS

Two clouds can share broad high-variance structure while differing in which points are closest. We therefore also compare each query's top-k neighbor sets, excluding the query itself. If o_k is their average fractional overlap over a population of n candidates, we report

$$\mathrm{N@}\,k = \frac{o_k - k/(n-1)}{1 - k/(n-1)}. \tag{8}$$

This maps the expected overlap of independent random neighbor sets to zero and identical neighborhoods to one (Huh et al., 2024).

The endpoint comparisons in Figure 1 rank neighbors by mean-centered dot product: subtract the mean of the compared token vectors before computing similarities. The contextual and prior-work comparisons rank by ordinary squared Euclidean distance, $D_{ij} = G_{ii} + G_{jj} - 2G_{ij}$. Neither ranking uses the HSIC correction. Appendix B.2 explains the additional reference-family normalization used only in Figure 1; for the other figures, one means identical neighbor sets.

A.3 FROM PREDICTIONS TO AN ANSWER REFERENCE

A logit vector specifies predictions, but its coordinates are vocabulary items rather than residual-stream directions. The pullback asks which state would express those predictions through a fixed readout. It therefore lets us compare hidden representations with a reference built from another model's predictions, or from known task targets, without borrowing that model's hidden states.

Write $\boldsymbol{W} \in \mathbb{R}^{V \times d}$ for the readout with one vocabulary token per row. When hidden states are expressed before the learned final normalization gain, we fold that gain into $\boldsymbol{W}$. Subtracting the mean vocabulary row gives $\overline{\boldsymbol{W}}$; likewise, we center target logits $\boldsymbol{z}$ over the same V tokens. This removes the additive constant that does not change softmax probabilities. The reference state is the minimum-norm least-squares solution $\boldsymbol{a} = \overline{\boldsymbol{W}}^{+}\boldsymbol{z}$. The regularized implementation is

$$\boldsymbol{a}_\lambda = \left(\overline{\boldsymbol{W}}^\top \overline{\boldsymbol{W}} + \lambda \boldsymbol{I}\right)^{-1} \overline{\boldsymbol{W}}^\top \boldsymbol{z}, \qquad K_{\text{answer}}(\boldsymbol{c}, \boldsymbol{c}') = \langle \boldsymbol{a}_\lambda(\boldsymbol{c}), \boldsymbol{a}_\lambda(\boldsymbol{c}') \rangle. \tag{9}$$

We do not normalize each pullback vector to unit length. The reference retains both the structure of the predictions and the geometry induced by the readout. Output support and regularization are specified with each experiment below.

For an affine readout $\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}$, subtract the fixed offset $\boldsymbol{b}$ before vocabulary centering to recover the state in the same coordinates. Retaining it instead adds a context-independent translation to every pullback, which leaves both centered CKA and ordinary-distance neighborhoods unchanged. This applies to Pythia's final LayerNorm bias.

For a model's own logits, $\boldsymbol{z} = \overline{\boldsymbol{W}}\boldsymbol{h}$, the unregularized pullback is the projection of $\boldsymbol{h}$ onto the row space of the centered readout. It equals $\boldsymbol{h}$ when the readout has full column rank. Thus the own-answer comparison tests how early final prediction geometry appears; independent predictions test whether a similar reference can be obtained without using the target model's final state.

B Shared Endpoint Geometry and Shallow Recovery

Figure 1 asks whether shallow models recover relationships between tokens that recur across much larger models. Here we specify which tokens and checkpoints are compared, how to read the reference-family baseline, and how the geometric comparison relates to prediction quality.

B.1 Matching tokens across model families

We match exact surface strings that encode and decode reversibly as one nonspecial token in every tokenizer. The primary population contains 9,635 GPT-2 tokens with FineWeb frequency rank below 10,000 that meet this criterion for base Llama-3.1-8B, Qwen2.5-7B, Pythia-2.8B, and OLMo-2-7B. Thus each kernel row refers to the same string even though its token ID differs across models. Leading spaces are part of the matched string.

At E we use the input lookup rows. At U we use LM-head rows after subtracting their full-vocabulary mean, without folding in the final normalization gain. This compares lexical weights; contextual pullbacks instead use the effective readout of Appendix A.3. Quantitative panels use ordinary dot-product kernels, not row-normalized cosine kernels.

B.2 The reference-family baseline and frequency controls

The consensus is the equal mean of the four unit-Frobenius kernels defined in Equation (1). To evaluate a reference model, we exclude its entire family from that mean. A shallow model is compared with all four families, since none belongs to its family. This prevents the reference models from receiving credit for agreement with themselves.

Panel A reports CKA directly. The frequency and neighborhood panels divide recovery by the mean leave-one-family-out agreement of the four reference models. One therefore means *agreement at the reference-family mean*, not perfect reconstruction or a theoretical ceiling. For neighborhoods, the chance correction is applied to both numerator and denominator before taking this ratio.

Neighborhood evaluation uses 819 fixed, frequency-stratified query tokens and all 9,635 tokens as candidates, with $k = 5, 10, 25, 50, 100$. We center vectors over this candidate population before ranking dot products, keeping the original full-Gram normalization factors fixed when averaging model kernels. Frequency-band CKA uses the fixed anchors within GPT-2/FineWeb rank bands 0–10k, 10k–20k, 20k–30k, 30k–40k, and 40k–50,257. The two checks ask whether agreement extends beyond frequent tokens and beyond broad global structure, respectively.

B.3 Training the shallow and deeper comparison models

All controlled models shown in Figure 1 use the GPT-2 tokenizer and the same 8B-token stream from `monology/pile-uncopyrighted`, a variant of the Pile (Gao et al., 2021). Their configurations are:

Model	Width	Heads	Microbatch tokens	MLP
0-layer	768	–	65,536	None
1-layer	768	12	65,536	None
16-layer	768	12	8,192	SwiGLU, width 2688

All use context length 512, untied input/output weights, RMSNorm, and BF16 forward computation. The zero-layer model has no attention, MLP, or positional contribution: it reads out the normalized token embedding. The one- and 16-layer models use learned position embeddings. The zero- and one-layer models have 77.6M and 79.9M parameters, respectively. All three models are width-matched; the 16-layer model also includes MLPs. The contextual comparisons in Figure 2 retain a width-1536 one-layer checkpoint trained on the same 8B-token stream.

Training uses AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.05, and a global batch of 524,288 tokens. The peak learning rate is 3×10^{-4} for matrix parameters and 6×10^{-4} for embedding lookups, with 64 warmup steps and cosine decay to one tenth of the peak. The displayed checkpoints use seed 0 and the same data ordering.

B.4 Geometric agreement despite a prediction-quality gap

To put geometric recovery in context, we score the same held-out Pile text prefix, spanning 999,936 GPT-2 prediction tokens. Each model retokenizes this text and is evaluated in reset blocks of at most 512 native tokens, without an added BOS. We report negative log likelihood in bits per UTF-8 byte, rather than comparing perplexities with different token denominators. Llama, Qwen, Pythia, and OLMo obtain 0.690, 0.705, 0.726, and 0.746 bits per byte; the one- and zero-layer models obtain 1.486 and 2.123. The same native-token context limit can span different amounts of text, but the large performance gap makes the central comparison clear: recovering shared geometry does not require comparable predictive capability.

B.5 Reading the calendar and integer examples

The qualitative panels isolate angular structure: subtract the group's mean vector, normalize rows, and center again before computing PCA or the Fourier spectrum. Month and weekday tokens have a leading space and match across all four families; their references average normalized group kernels before projecting the consensus. The years and integer panels instead compare directly with Pythia-2.8B. Years use leading-space tokens 1900–1999 at E; the other three families split these strings into multiple tokens. The integer spectrum uses bare tokens 0–99 at E. PCAs are computed independently and aligned by rotation/reflection for display. The integer spectrum evaluates complex Fourier modes 1–50 and normalizes their energies by the sum over those modes. These examples illustrate the quantitative agreement; neither their ordering nor their 2D coordinates enters the CKA or neighbor comparisons in panels A–D.

C Input and Answer Geometry Inside the Network

Figure 2 compares hidden states with fixed endpoint references. It asks how far input geometry persists and how early prediction geometry appears, and whether shallow-model interfaces provide similar references. This section explains the four curves and checks what their agreement depends on.

C.1 FineWeb contexts and hidden states

We start with 2,048 FineWeb prefixes of 128 Llama tokens and feed the same decoded text to base Llama-3.1-8B, Qwen2.5-7B, Pythia-2.8B, and OLMo-2-7B, without chat wrappers. We retain the 2,041 contexts whose final token under every native tokenizer maps to an exact single GPT-2 token, giving a common population for the one-layer embedding comparison. Each target's input references follow its own final token. We capture the final-position residual after each block: 32 for Llama, Pythia, and OLMo, and 28 for Qwen.

To compare directions without changes in residual magnitude dominating the result, hidden states and current-token embeddings receive scalar RMS normalization, $\rho_m(\boldsymbol{h}) = \boldsymbol{h}/\sqrt{\|\boldsymbol{h}\|^2/d_m + \epsilon_m}$, using the checkpoint's epsilon. For RMSNorm models, final states are expressed in these gain-free coordinates. Pythia's final LayerNorm also subtracts the coordinate mean; its final states use those gain- and bias-free LayerNorm coordinates. The learned final gain is folded into each readout. Pythia's fixed LayerNorm bias induces a common translation in its pullback and does not affect either metric (Appendix A.3). The one-layer input reference uses the same transformation at width

1536 with $\epsilon = 10^{-6}$. It uses the token lookup vector, not a contextual forward pass or a positional embedding.

C.2 What the four reference curves measure

Curve	Reference representation for each context
Shape of the input	The target model's current-token embedding.
1-layer embeddings	The matching token's embedding in the frozen one-layer model.
Shape of the answer	The target model's own centered final logits pulled back through its effective unembedding.
1-layer pullback	Mistral-Nemo Base 12B logits pulled back through the frozen one-layer model's effective unembedding.

The one-layer checkpoint is the width-1536, 8B-Pile-token model described in Appendix B.3. Its two references test complementary claims: whether its token relationships resemble those expressed in early hidden states, and whether an independently supplied prediction can acquire useful late-layer geometry through its output interface. None of these references is fitted to the target model's intermediate states.

Own-answer reference. We center over the selected output support and fold the final gain into each target's unembedding. Llama and Qwen use 62,787 exact output-token surfaces shared with Nemo; OLMo uses the 55,959-token intersection shared by those three models and OLMo. Pythia uses its full native vocabulary, including the fixed readout offset from its final LayerNorm bias. The ridge pullback in Equation (9) uses $\lambda = 10^{-5}$ times the mean diagonal of the interface Gram matrix. A fixed 32,768-dimensional CountSketch (Charikar et al., 2002) approximates the normal-equation products for Llama, Qwen, and OLMo; Pythia's products are computed exactly. Relative reconstruction errors are below 8×10^{-4} for all four models, after removing the common translation for Pythia. This makes the own-answer curve an endpoint-persistence test: the informative observation is agreement before the final readout.

Independent prediction through a shallow interface. For the one-layer pullback, we intersect the same output support with the GPT-2 vocabulary, leaving 34,205 exact shared tokens. Nemo supplies the centered logits on this support; the one-layer model supplies the gain-folded unembedding. We solve the normal equations directly in float64, using the same relative ridge and no sketch or post-pullback normalization. The resulting reference is identical for all four target models. Changing the ridge by a factor of ten in either direction changes it by less than 0.05% in relative norm. The other curves retain their original support.

This comparison separates prediction from the computation that produces it: the shallow model need not generate Nemo's predictions to express their geometry through its interface.

C.3 Layerwise comparison and model averaging

For each layer and reference we compute CKA and ordinary-distance neighbor overlap directly. There is no kernel mixture, learned coefficient, or allocation of agreement into colored components. We use three document-disjoint half partitions and their reversals, yielding six evaluation halves. CKA is averaged equally across halves; neighborhood overlap is weighted by the number of queries. Neighbors are drawn within each evaluation half, with self-neighbors excluded and seeded tie-breaking for repeated input tokens.

We express depth as the fraction of completed blocks and interpolate each model's scores onto the common interior-depth grid, without extrapolation. The main curves are the equal mean of the four models; their bands span the minimum and maximum across models, not a confidence interval. First and final endpoints are omitted from the main figure to emphasize agreement inside the network. These are mean *within-model reference agreements*, not a decomposition of cross-model CKA or fractions of explained activation variance.

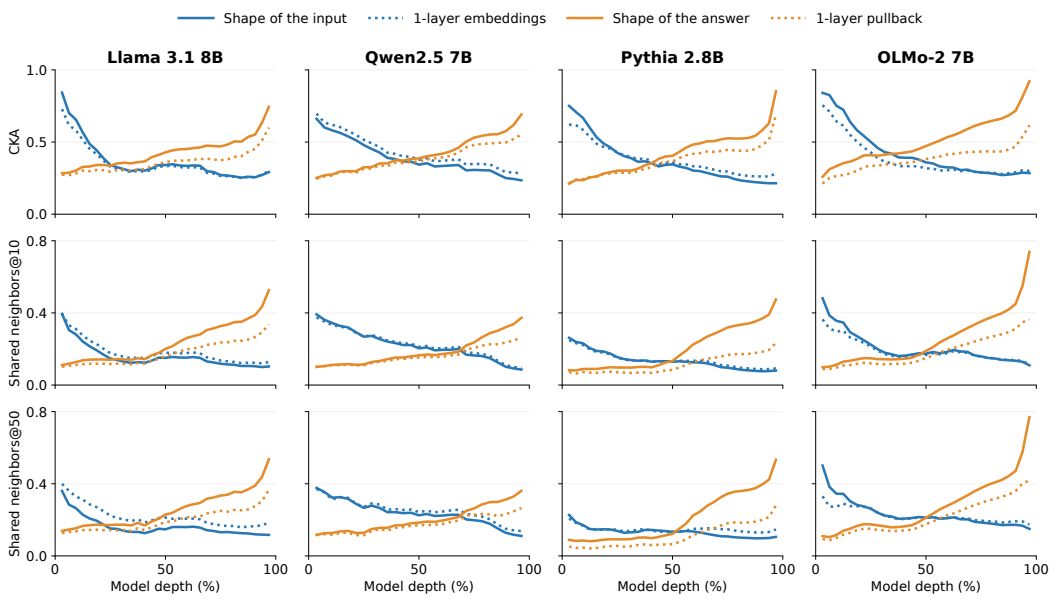


Figure 6: **Endpoint agreement within each model and at two neighborhood sizes.** Columns show Llama, Qwen, Pythia, and OLMo; rows show CKA, neighbors@10, and neighbors@50 on the same 2,041 contexts. The four references are those of Figure 2. Scores use native model depths and direct reference–state comparisons, with no fitted combination of kernels.

C.4 Does the transition survive model and neighborhood choices?

Figure 6 separates the four models and adds neighbors@50 alongside the main neighbors@10 measure. This makes it possible to distinguish a consistent trend from one created by averaging, and broad agreement from overlap among only the closest contexts. The four references and their normalization are unchanged. At the last displayed depth, the one-layer pullback reaches mean neighbor overlap 0.296 at $k = 10$ and 0.328 at $k = 50$: the independent reference preserves local relationships as well as global alignment.

C.5 What token identity and output correspondence contribute

Token-identity baseline in the main figure. For each model, we assign each distinct current-token ID an independent Gaussian vector of the native embedding width, sampled from $\mathcal{N}(0, \boldsymbol{I}/d_m)$ and given the same scalar RMS normalization. The dictionary is fixed across prompts, layers, and evaluation splits: repeated tokens always receive the same vector. We average scores over 32 dictionaries, then over the four families on the same 2,041 contexts as Figure 2. Bands span the family means, not dictionary uncertainty. At the first displayed depth, learned embeddings versus this baseline yield CKA 0.767 versus 0.495 and neighbors@10 0.375 versus 0.167. The baseline retains token identity without systematic learned relationships between different tokens; it is not an identity matrix or a permutation of contexts. Hidden states and answer references are unchanged.

Additional two-model controls. These auxiliary controls use Llama and Qwen on the original 2,048-context population, before the additional seven-context filter for the four-family comparison. They are not four-model averages.

Repeated tokens already create input structure. We replace each target's embedding dictionary with independent Gaussian vectors at its native width, preserving token identities and applying the same scalar RMS normalization. Repeated occurrences of a token still share a vector. Averaging nine dictionaries, this reference reaches CKA 0.495 at the first displayed depth, compared with 0.741 for learned embeddings; neighbors@10 rises from 0.165 to 0.386 when learned embeddings are used (Figure 7). Thus occurrence patterns explain a substantial baseline, while relationships between



Figure 7: **Input identity and prediction structure provide informative baselines.** Learned current-token embeddings are compared with random dictionaries retaining token identity; the one-layer prediction pullback is compared with a token-row-permuted interface. Lines average Llama and Qwen after averaging nine randomizations within each model. CKA and neighbors@10 are computed directly on the original 2,048-context Llama–Qwen population.

different tokens add both global and local agreement. This is a direct reference comparison, not an ablation of the transformer or a fitted attribution.

Prediction structure survives a scrambled interface. We keep Nemo's logits fixed and randomly permute the token rows of the one-layer unembedding on the same 34,205-token support. This preserves its singular values and conditioning while breaking token-to-readout correspondence. At the last displayed depth, the learned versus permuted pullbacks have CKA 0.571 versus 0.517 and neighbors@10 0.298 versus 0.282, averaging nine permutations. Much of the alignment therefore survives without the correct output-token correspondence. On these natural contexts, structure in the predictions is an important contributor; the additional gain from the correctly aligned interface is smaller. The symmetric weekday targets in Figure 4 isolate the interface's contribution more directly because those targets have no preferred cyclic order.

C.6 Wording can dominate geometry while answer identity remains readable

The endpoint comparisons concern entire similarity matrices. A complementary experiment asks which of two prompts is closer: one with the same wording but a different answer, or a paraphrase with the same answer?

We cross eight ParaRel relations (Elazar et al., 2021), eight wordings per relation, and 24 facts with distinct single-token answers, giving 1,536 prompts. Each uses a question line `Q: Fill in the blank: [pattern]` followed by `A:` on a new line; for example, "[X] was founded in [Y]" versus "[X] started in [Y]", with the subject inserted and the answer left blank. Within each wording, token length is matched across facts in base Llama and Qwen. We compare cosine similarities after centering the 192 states within each relation, averaging all same-answer versus same-wording comparisons equally. Bands bootstrap the eight complete relation trajectories, using 4,000 resamples.

All eight relations favor wording after the first block and answer identity at the final block in both models. Aggregate same-answer win rates rise from 0.107 to 0.919 for Llama and from 0.294 to 0.890 for Qwen. Yet a nearest-prototype readout trained on seven wordings identifies the answer label in the eighth after the first block with 86% and 98% accuracy, against $1/24$ chance. Because each fact has a unique answer, this readout can use subject identity; it measures accessible answer-label information, not successful answer computation. The distinction is useful: information can be readable even when another factor dominates similarity.

D Previously Reported Geometries

Table 1 links reported arrangements to the shape of the input or the answer. Each subsection below identifies the observation being explained, the corresponding reference, and the evidence connecting them. The SAE comparison uses published feature clusters; the other hidden-state comparisons use

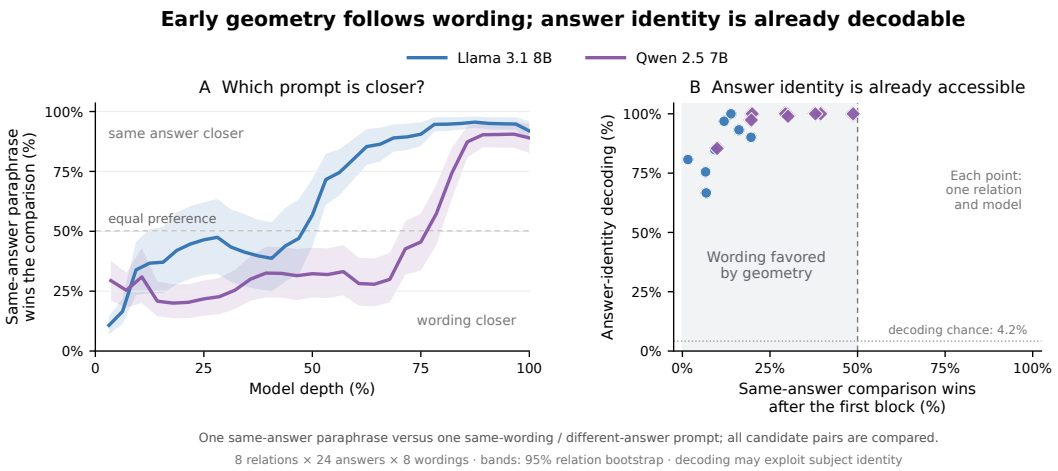


Figure 8: **Geometric prominence and readable information differ.** Left: probability that a same-answer paraphrase is closer than a same-wording prompt with a different answer; 50% is equal preference. Right: answer-label readout across unseen wordings after the first block. Readout transfers across wordings of the same facts, not to unseen facts.

our own captures. Global scores use corrected CKA and local scores use ordinary-distance neighbors as defined in Appendices A.1 and A.2. PCA illustrates the arrangements, while the quantitative comparisons specify explicitly whether they concern the full representations or a selected PC plane.

D.1 Calendar structure in GPT-2 and Mistral-7B

Engels et al. (2025) report calendar structure at both ends of the network. On the input side, clustering sparse-autoencoder (SAE) features of GPT-2 at the residual before block 7 recovers circles for weekdays and months and an ordered arc for the years of the twentieth century. Mistral-7B layer-8 clusters add a *weekend* point between Saturday and Sunday and seasons between their months. On the answer side, final-token states of Mistral-7B in their modular-arithmetic prompts contain a circle in the *target* day, visible in layers 17–29 once one-hot start-day and duration effects are regressed out (their Fig. 23, layer 25). We apply the persistence test of Section 5 to both: input-side structure should already agree with the token embedding at every layer up to the one studied, and answer-side structure should agree with the shape of the answer at every layer from the one studied to the output.

Input side: GPT-2, every layer up to layer 7. Each weekday, month, and year from 1900 to 1999 (all single GPT-2 tokens) ends eight wordings, four ordinary ("The current month is January") and four arithmetic ("What month is three months before January"). We read the residual stream at that word before every block and after the last, and compare it with the word's own embedding row. The released SAE clusters for the three families are reconstructed at layer 7 as in the original work.

Agreement with the embedding holds at every layer from the embedding to layer 7 (Figure 9): CKA is 0.97–1.00 for weekdays, 0.93–1.00 for months, and 0.97–1.00 for years, and shared neighbors@2 is 0.89–1.00, 0.86–1.00, and 0.80–0.99. The permutation 95th percentiles are at most 0.16 (CKA) and 0.13 (neighbors). The published SAE reconstructions agree with the embedding to a similar degree (CKA 0.97, 0.94, 0.91). Agreement also continues after layer 7 and declines only modestly by the output (months: CKA 0.85 after the last block). The calendar arrangements therefore never have to be constructed within the first seven blocks: they are present in the embedding and preserved along the residual path. This is consistent with GPT-2 carrying these circles while solving almost none of the corresponding arithmetic prompts (8/49 weekdays and 10/144 months; Engels et al., 2025).

Weekend and seasons in the input embedding. We test the additional Mistral-7B points directly in the input embeddings of Mistral-7B, GPT-2, and Llama-3.1-8B, using lowercase single tokens and scalar-RMS Euclidean distances in the full embedding space (Table 2). In all three models, the two days nearest to *weekend* are Saturday and Sunday, which 6–9% of 5,000 random word-like tokens achieve. The nearest month to every season word lies within that season in all fifteen cases (random

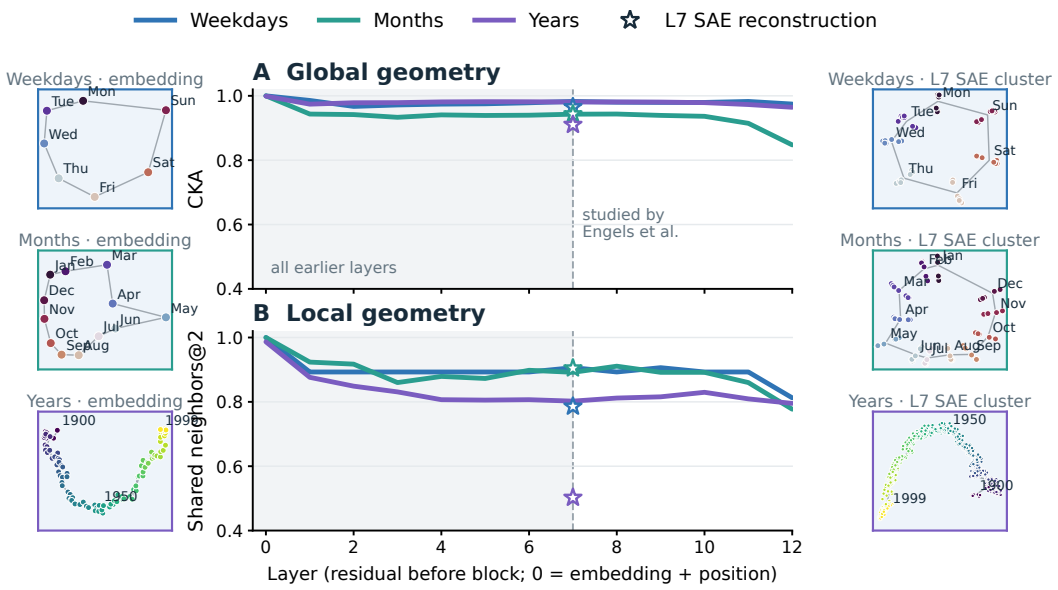


Figure 9: **Calendar geometry in GPT-2 is present in the embedding and at every layer before the one studied by Engels et al. (2025). A:** CKA and **B:** chance-corrected shared neighbors@2 between states at the calendar token and that token's embedding, averaged over eight wordings. Layer 0 is the embedding plus position; the dashed line marks layer 7. Stars mark the published layer-7 SAE-cluster reconstructions. Label-permutation 95th percentiles are at most 0.16 (CKA) and 0.13 (neighbors). Left: embedding PC1/2. Right: PC1/2 of the SAE reconstructions after centering each wording, matching the within-wording metrics.

tokens: 10–46% per season). The second-nearest month is occasionally the adjacent month outside it, and *fall* is the only word with a distant second neighbor, possibly due to its ambiguity.

Table 2: Nearest calendar words to *weekend* and the seasons in three input embeddings (scalar-RMS Euclidean distance, full dimension).

Word	Mistral-7B	GPT-2	Llama-3.1-8B
weekend	Sat, Sun	Sat, Sun	Sat, Sun
winter	Dec, Nov	Dec, Feb	Dec, Nov
spring	Mar, Apr	Mar, Feb	Apr, Mar
summer	Jul, Jun	Jul, Jun	Jul, Aug
autumn	Oct, Nov	Oct, Aug	Oct, Nov
fall	Oct, May	Nov, Oct	Sep, Oct

Answer side: Mistral-7B, every layer from layer 25. We use the released prompts and scoring of Engels et al. (2025) ("Let's do some days of the week math. Two days from Monday is", 49 weekday and 144 month prompts, scored by the highest-logit valid answer); Mistral-7B answers 30/49 and 125/144 (reported: 31/49 and 125/144). The reference for the shape of the answer is an equal-margin logit vector on the correct day or month, pulled back through the centered, gain-folded unembedding with the relative ridge of Appendix C.1. These targets place every answer at the same distance from every other, so any cyclic order in their pullback comes from the unembedding. Following the original analysis, one-hot start and duration effects are regressed out of both the final-token states and the reference before comparison.

This regression is needed because the duration embedding dominates the final-token geometry through most of the network (CKA up to 0.78 for weekdays and 0.69 for months). Once it is removed, agreement with the shape of the answer rises after layer 17 and holds at every layer from 25 to the output (Figure 10): CKA is 0.74–0.77 for weekdays and 0.68–0.81 for months, compared with at most 0.22 before layer 17, and 0.45–0.55 and 0.57–0.84 of each prompt's nearest neighbors share its

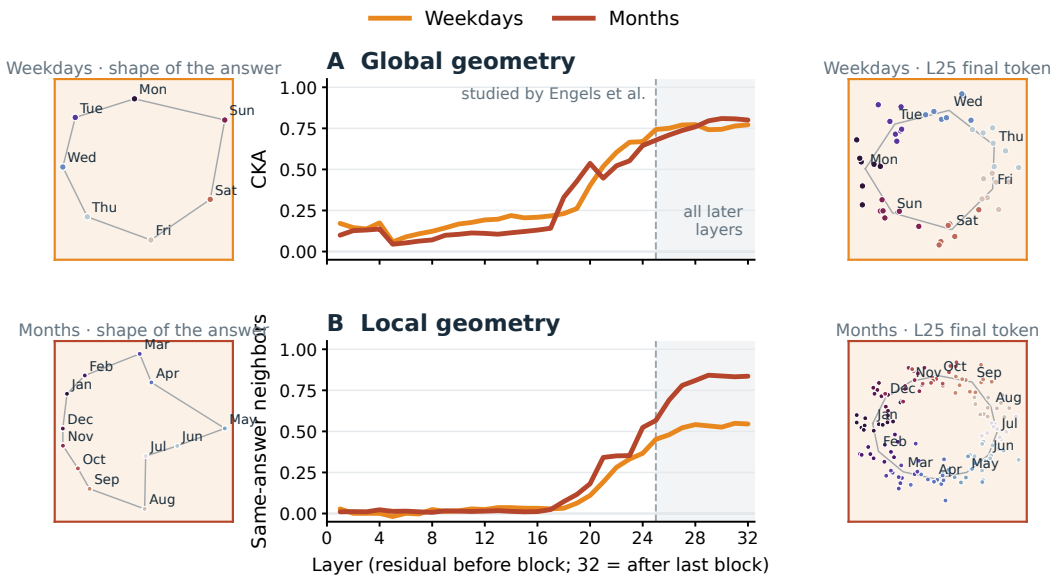


Figure 10: **The late answer circle in Mistral-7B matches the shape of the answer at every layer from the one studied by Engels et al. (2025).** Final-token states in their weekday and month prompts are compared with equal-margin answer targets pulled back through the unembedding, after regressing one-hot start and duration effects out of both. **A:** CKA. **B:** chance-corrected fraction of each prompt's nearest neighbors that share its answer, with k equal to the number of other prompts with the same answer (6 for weekdays, 11 for months). The dashed line marks layer 25. Left: PC1/2 of the answer pullback, whose targets are equidistant, so its cyclic order comes from the unembedding. Right: PC1/2 of the layer-25 states, colored by the correct answer.

answer after chance correction, against permutation 95th percentiles of at most 0.06. The late circle in the target therefore matches the pullback of symmetric answer targets through the model's own readout. Mistral's errors are almost all off by one in the backward direction, so predicted and correct answers are neighbors on the circle and cannot be separated in this task.

D.2 Periodic number structure

Kantamneni & Tegmark (2025) find that GPT-J, Pythia-6.9B, and Llama-3.1-8B represent integers as a generalized helix, with Fourier periods $T = 2, 5, 10, 100$ and a linear component, fitted at the operand tokens after layer 0. For addition prompts they report a second helix, of the sum, at the last token: MLPs 14–18 of GPT-J construct it and later layers read it out to the logits from layer 19 onward. They attribute the periodic features to their usefulness for arithmetic and conjecture that they serve as an error-correcting code (their App. E). We apply the persistence test to both helices in GPT-J, their main model, and treat the origin of the periodic token structure separately.

Origin of the periodic structure. The integer embeddings of the one-layer model already show the same period-ten family of Fourier peaks as Pythia-2.8B (Figure 1, bottom), although this model cannot perform multi-step arithmetic. The tokenizer intervention in Figure 5 reduces period-ten energy in both shallow and deeper models by changing which tokens follow integers, with no arithmetic training signal. Combined, these results suggest that Fourier this specific fourier structure in hidden representations depends more on low level statistics than on arithmetic computation.

Input side: the operand helix. We use their prompt for GPT-J, "Output ONLY a number. {a}+{b}=", for all $a, b \in \{0, \ldots, 99\}$; GPT-J answers 84.1% correctly (reported: 80.5%). The state at the operand token `a` depends only on the preceding text, so each layer contributes 100 states, compared with the operand's embedding row. GPT-J uses LayerNorm, so states are put in its gain-free coordinates (feature mean removed, then rescaled). Agreement with the embedding holds at every layer (Figure 11): CKA is 0.85–0.95 and shared neighbors@2 is 0.45–0.72 after the embedding, against label-permutation 95th percentiles of 0.03. Period-ten energy relative to an even spectrum is

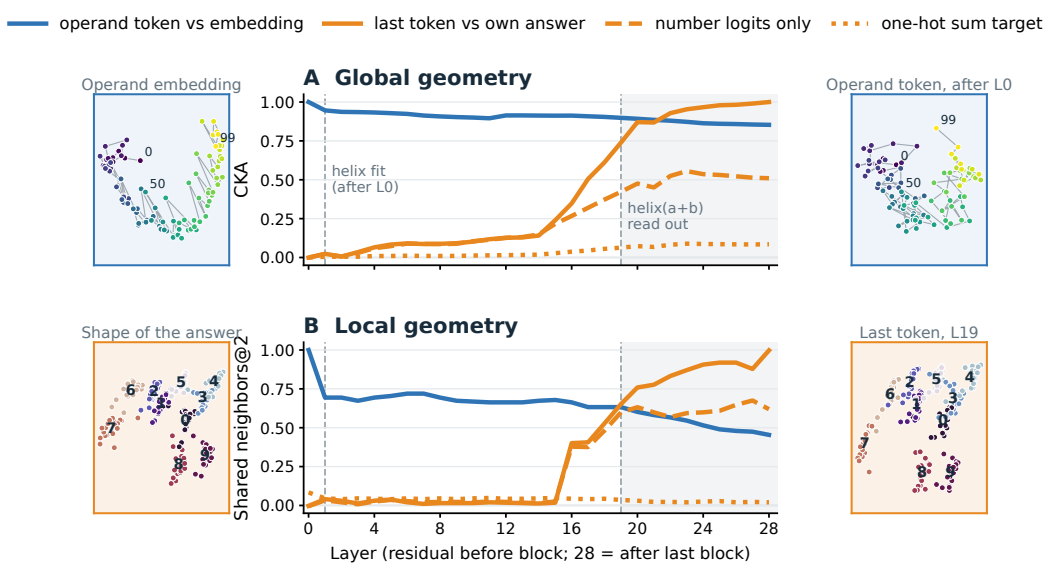


Figure 11: **GPT-J's number helices match the shape of the input and the shape of the answer.** Addition prompts of Kantamneni & Tegmark (2025). **A:** CKA and **B:** chance-corrected shared neighbors@2 across layers. Blue: the state at the operand token against that operand's embedding. Orange: the last-token state against the shape of the answer after regressing one-hot operand effects out of both sides, using GPT-J's own logits (solid), only its logits on the 199 sum tokens (dashed), or a one-hot target on the correct sum (dotted); neighbors use the 199 sum centroids. Dashed lines mark where the operand helix is fitted (after layer 0) and where the sum helix is read out (layer 19). Left: references (embedding PC1/2 colored by value; answer PC2/3 colored by the units digit of the sum). Right: states projected onto the reference's principal plane, with nothing fitted to them (operand token after layer 0; last token at layer 19).

already 1.68 in the embedding and grows to 2.69 after layer 0, where the helix is fitted, and 3.41 at the output. The operand helix is therefore present in the embedding and carried along the residual path rather than constructed for the task.

Answer side: the sum helix. The shape of the answer is GPT-J's own centered logits at the last token pulled back through its centered, gain-folded unembedding, as in Section 4. The pullback recovers the final state up to a relative error of 0.03. As in Appendix D.1, one-hot operand effects are regressed out of both sides, since the operand embeddings dominate last-token geometry through the middle layers (CKA up to 0.46 for a and 0.69 for b). Local agreement is measured on the 199 sum centroids. Agreement with the shape of the answer stays below 0.15 through layer 14, rises while MLPs 14–18 build the sum helix, and holds at every layer from layer 19 to the output: CKA is 0.74 at layer 19 and at least 0.87 from layer 20, and shared neighbors@2 is 0.65 and at least 0.76, against permutation 95th percentiles of at most 0.02. Restricting the logits to the 199 sum tokens retains about half of this agreement (CKA 0.42–0.56, neighbors 0.57–0.68 from layer 19). Unlike the weekday task of Figure 10 part of the structure comes from the specific distribution predicted by the model, not only the Fourier structure in the unembedding weights.

D.3 Calendar answer manifolds

Wurgaft et al. (2026) study answer manifolds in Llama-3.1-8B on prompts such as "What is four months after January?", averaging activations at their layer 28 over prompts that share an answer. These activation centroids are nearly isometric to the model's output distributions over the answer concepts, and steering along them produces smooth transitions in behavior. The authors leave the origin of this shared geometry open ("we do not study the origins of the shared geometry between behavior and representation") and argue that data statistics cannot account for it in their in-context tasks. We apply the persistence test on the answer side: the arrangement should agree with the shape of the answer at every layer from the one studied to the output.

Setup. We use their weekday, month, letter, and age prompt families (their Table 1) and a second wording of each. Their layer 28 is the residual after zero-based block 28, L29 in our indexing. The shape of the answer is the model's own centered logits pulled back through its effective unembedding. As in Appendices D.1 and D.2, one-hot source and increment effects are regressed out of both the scalar-RMS states and the pullback within each wording. Without this step, answer centroids already agree with the answer reference in early layers (for weekdays, CKA reaches 0.96 by layer 16), because on a balanced source-by-increment grid, averaging input-dominated states by answer yields centroids ordered like the answers themselves. Scores compare individual prompts and average the two wordings; 200 label permutations per wording give the null.

Results. After the regression, agreement is low through the middle of the network, rises between layers 17 and 24, and holds at every layer from L29 to the output (Figure 12). At L29, CKA is 0.98 for weekdays, 0.97 for months, and 0.90 for ages, and shared neighbors@2 is 0.84, 0.87, and 0.81, compared with at most 0.45 and 0.08 through layer 16 and permutation 95th percentiles of at most 0.06 and 0.04. Letters are shown in neither panel: with only two increments, removing source and increment effects leaves too few degrees of freedom for a per-layer test (their L29 CKA is 0.85, but neighbors@2 is already 0.92 at layer 1, against a null of 0.39).

Why the isometry is expected. At the layer they study, the activation manifold is the shape of the answer. Near the output, each state is approximately the pullback of its own centered logits, so distances between answer centroids are distances between the model's predictions as expressed through the unembedding. The behavior manifold of Wurgaft et al. (2026) is an order-preserving transformation of the same predictions, their restriction to the answer concepts followed by a square root. The two manifolds are therefore two views of the model's predictions, and their near-isometry follows from the late states carrying the answer, without requiring a separate representation of the concept from which both are derived. The same holds in their in-context graph tasks, where answer geometry follows whatever the model predicts in context (Appendix D.6). Their "pullback" geometry, a metric on activation space induced by the Jacobian of the map to outputs, differs from the pullback of logits through the unembedding used here.

D.4 Belief-state geometry from structured prediction uncertainty

In Mess3, a history leaves uncertainty about the process's hidden state and next symbol. Shai et al. (2024) already connect the resulting belief geometry to prediction. This makes Mess3 a useful instance of the *prediction* side of Figure 4: the target distribution itself has rich geometry, independently of the neural network.

We evaluate our four-block, width-64 model on all $3^9 = 19,683$ length-nine histories, weighted uniformly. Bayes-optimal next-token probabilities come from the known process. We center their logarithms over the three symbols and pull them back through the centered unembedding. The comparison uses raw residuals after blocks 3 and 4, before final LayerNorm. The model and process parameters differ from the original authors' experiment.

Across three random half splits, the pullback reaches CKA 0.936 at L3 and 0.929 at L4, with ordinary-distance N@50 0.375 and 0.377. Each split uses 819 fixed queries and all held-out histories as neighbor candidates. Holding out complete three-symbol suffix groups yields CKA $0.919/0.909$ and N@50 $0.296/0.299$, so the match extends beyond histories sharing the same short suffixes with the other half.

Last-token identity has substantial CKA, about 0.80, but N@50 only about 0.01: it captures the coarse partition into three symbol groups without their fine structure. The oracle pullback resolves much more of that local organization. Oracle logits themselves have slightly higher global CKA than their pullback, consistent with this being primarily an example of structured predictive demands. The RRXOR task in the same paper distinguishes beliefs with identical immediate predictions using concatenated layers; it is outside this next-token comparison.

D.5 Weekday retrieval with symmetric answer targets

To isolate structure supplied by the interface rather than the target arrangement, Figure 4B uses seven equal-margin weekday targets. We compare them with base Llama-3.1-8B on the prompt family

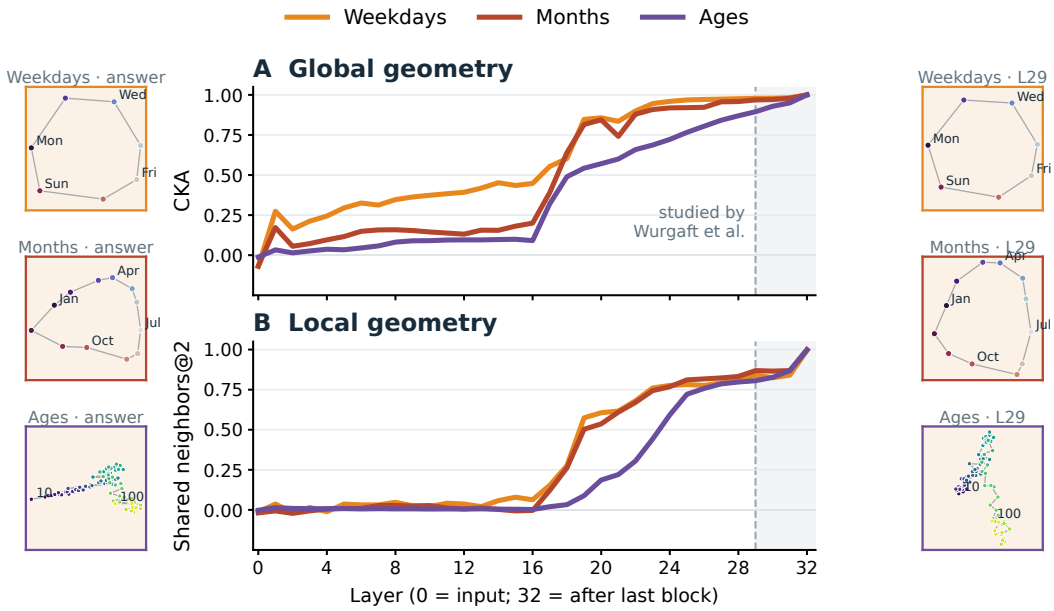


Figure 12: **Answer manifolds in Llama-3.1-8B match the shape of the answer at every layer from the one studied by Wurgaft et al. (2026).** Prompt families of Wurgaft et al. (2026) and a second wording. **A:** CKA and **B:** chance-corrected shared neighbors@2 between individual final-token states and the model's own logit pullback, after regressing one-hot source and increment effects out of both within each wording. The dashed line marks their layer 28 (L29 here; 0 is the input). Left: PC1/2 of the answer centroids of the pullback. Right: L29 answer centroids projected onto that plane, with nothing fitted to them. Letters are omitted because two increments leave too few degrees of freedom after the regression.

```
Answer in one word.
The week starts on Sunday.
Q: What is the first day of the week?
A:
```

varying `first` through `seventh` and measuring the final colon, without a trailing space. Greedy next-token answers follow the Sunday-first convention on six of seven prompts; the fifth-day question returns Friday rather than Thursday. Reference labels use the correct days.

For correct token y, prescribe the full-vocabulary centered logit vector $\boldsymbol{z}_y = \boldsymbol{e}_y - \mathbf{1}/V$. Every distinct pair has squared distance two, so there is no preferred calendar ordering. These are finite equal-margin logits, not one-hot probabilities. Their least-squares pullback through the centered unembedding acquires a cyclic component. Only 2.68% of target energy lies in the readout's range: the displayed pullback represents the attainable projection of these symmetric targets.

The pullback agrees with final normalized states at L32 and raw penultimate states at L31, with CKA 0.948 and 0.915. Both share six sevenths of their two nearest neighbors with it (chance-corrected N@2 0.786). Exact enumeration of all 7! label assignments gives permutation $p = 1/5040$ and $4/5040$. L31 was fixed in advance. Each PCA is fitted independently and aligned only for display. Because L32 already includes the final normalization gain, its readout does not fold in that gain again.

The changing ordinal inputs are also structured, but agree less strongly with final states (CKA 0.713). The example demonstrates that a structured interface can turn symmetric answer targets into an arrangement resembling late representations, without an arithmetic task being required.

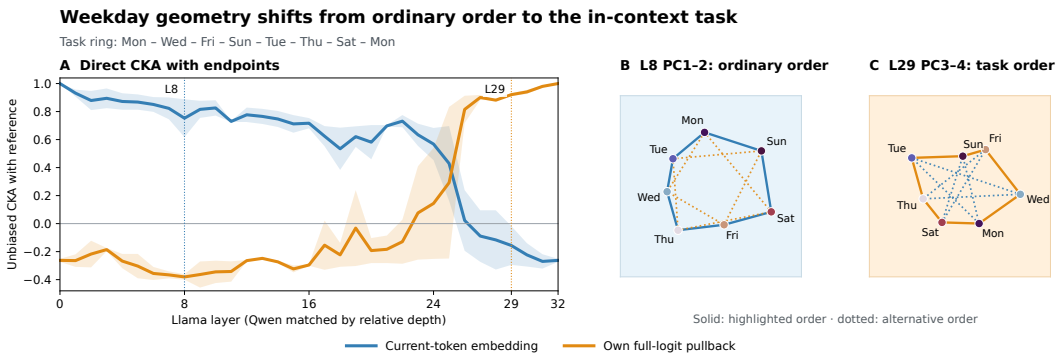


Figure 13: **Ordinary weekday geometry persists before task order emerges. A:** Direct corrected CKA with current-token embeddings (blue) and each model's own full-logit pullback (orange). Lines average Llama and Qwen; bands span their scores. With seven centroids, corrected CKA can be negative. **B–C:** Llama L8 PC1/2 shows ordinary weekday order, while L29 PC3/4 shows the context-defined ring. Solid edges highlight the corresponding order; dotted edges show the alternative. PCA bases are fitted on four context draws and points are shown on four disjoint draws. These PC views illustrate the two orderings; the CKA curves use all representation dimensions. Layers count completed blocks, so L22 is after zero-based block 21. This is a Park-style setting, not an exact replication of their serialization or layer choice.

D.6 Persistent weekday order under in-context graphs

Park et al. (2025a) highlight a counter-intuitive observation: ordinary weekday order persists in leading PCs even when context specifies a different ring. Since states are measured at the weekday tokens themselves, the input embedding supplies a natural reference for that persistent order.

Our related setup uses base Llama-3.1-8B and Qwen2.5-7B with 800 independently sampled, newline-separated neighbor pairs on the permuted ring Monday–Wednesday–Friday–Sunday–Tuesday–Thursday–Saturday–Monday. For each of eight context draws, we average states at source-weekday positions in the final 200 weekday observations into seven centroids. We average these centroids over each of two disjoint four-draw halves, apply scalar RMS normalization, and compare their hidden-state kernels directly with the model's own current-token embeddings and full-vocabulary final-logit pullback. The pullback uses the effective unembedding and gain-free normalized final states. No reference coefficients are fitted. Scores average the two halves, then the two models; Qwen layers are matched to Llama by relative depth. Both models assign more than 0.98 next-token probability mass to valid graph neighbors in this serialization.

At L22, current-token geometry remains strongly aligned with the full hidden-state kernel (mean CKA 0.732 across the two models). Answer alignment becomes stronger later, reaching 0.921 at L29. Together with the PCA views, this supports an inherited input component and a later answer-related component. It explains why familiar geometry can remain prominent while the model follows a new graph, without specifying how that graph is computed.

D.7 Calendar arithmetic: linking the two endpoints

Figure 3 brings the input and answer references together within one task. We cross 12 source months with offsets spelled out from one through twelve, giving 144 prompts in base Llama-3.1-8B:

> Q: What month is [offset word] months after [source month]?
> A:

We measure the final-colon residual after each block and apply scalar RMS normalization without the learned gain. The final state uses the same gain-free coordinates. Layer labels in this figure index block outputs from zero.

The two input references are the scalar-RMS-normalized embedding vectors of the offset and source-month tokens. They are fixed across depth: unlike our earlier routed-input analysis, they contain

no layer-dependent attention, Jacobian, or fitted gain. The answer reference is the unregularized full-vocabulary pullback of the model's own centered final logits through its effective unembedding. Because this readout has full column rank, the pullback equals the final gain-free normalized state. Agreement at earlier layers is the quantity of interest; agreement at the final endpoint is therefore one by construction.

For each reference separately, we compute corrected CKA against the hidden states and chance-corrected shared neighbors@10 using ordinary Euclidean distances. To avoid letting the repeated grid structure dominate a single score, we divide the prompts into three balanced, disjoint subsets of 48 using $(s + 2o) \bmod 3$ and average the three scores. These subsets are evaluation partitions, not train/test folds: the analysis fits no coefficients. Each local comparison has 47 candidate neighbors, so its chance overlap is $10/47$.

The *offset-identity baseline* uses the random-dictionary construction in Appendix C.5 at Llama's width, assigning one fixed vector to each of the twelve offset tokens across all 144 prompts. We average scores over 32 dictionaries; the faint band spans their pointwise central 95% range. At L5 this baseline yields CKA 0.515 and neighbors@10 0.255, below the learned-offset values 0.761 and 0.493. This compares learned offset relationships with repeated-offset identity, without changing the model or fitting a mixture of references.

The local and global metrics agree on the transition. At L5, offset geometry reaches CKA 0.761 and corrected neighbors@10 0.493, compared with 0.231 and 0.131 for answer geometry. At L22, the ordering reverses: answer geometry reaches CKA 0.837 and neighbors@10 0.670, compared with 0.213 and 0.107 for the offset reference. The answer curve first exceeds the offset curve at L19 under both metrics. Results for neighborhoods@5 and neighborhoods@20 are stored with the figure analysis and show the same progression.

Panels C–G use ordinary PCA of all 144 individual scalar-RMS states, without residualizing source month or offset and without averaging before fitting the PCA. Larger markers show the 12 group centroids after projection. Panels C–E show PC1/PC2 at L5, L10, and L18 and color prompts by offset; panels F–G show PC1/PC3 at L22 and L29 and color them by answer month. The lines merely join group centroids to make their ordering legible; they are not trajectories.

E Training Statistics Shape the Interfaces

Figure 5 tests whether changing token-level prediction statistics changes geometry in the expected direction. The raw text is preserved, but the tokenizer changes which prediction problems the model encounters. We first target a particular numerical periodicity, then test broader pairwise changes. A final control separates continuation-side from target-side training pressure.

E.1 Numeric tokenization and period-ten integer structure

Which predictive relationship changes? Integers sharing a last digit often share an ordinal continuation: for example, both `21` and `31` can precede `st`. We alter numeric token boundaries so that these recurring suffix predictions no longer attach to the same integer tokens. The standard GPT-2 vocabulary gives irregular boundaries to long integers. We add 100 continuation tokens, one for each two-digit string, and 13 tokens for valid ordinal endings. ASCII digit spans beginning after a literal space are then represented by an existing one- or two-digit leading token followed by right-aligned two-digit chunks; valid ordinals receive a dedicated terminal. For example, `␣5000` changes from one token to `␣50|00`, and `␣21st` changes from `␣21|st` to `␣2|1st`. The transformation is exactly reversible and leaves the tokens for leading-space integers 0–99 unchanged, so their representations can be compared directly.

On the 10M-token held-out Pile stream, we form conditional-Hellinger rows $\sqrt{p(\cdot \mid n)}$ over the full vocabulary for leading-space integers $n \in \{0, \ldots, 99\}$. Numeric chunking reduces their absolute $k = 10$ energy to 0.653 of the original value, and the excess similarity between integers with the same final digit falls from 0.0189 to 0.0100.

Matched one- and eight-layer models. Within each depth, the baseline and intervention use the same seed, data order, optimizer, width 768, 12 heads, context length 512, and 2B-token training

budget. The shallow model has one attention layer and no MLP; the deeper model has eight attention-and-SwiGLU blocks. The expanded vocabulary contains 50,370 rows, adding 113 independently initialized rows while copying every shared parameter exactly from the matched baseline initialization. Retokenizing the 2B-source-token cache produces 0.34% more tokens, so the fixed 2B transformed-token budget covers slightly less raw text. Figure 5C reports one paired seed at each depth.

Measuring the repeating pattern. For a site matrix $\boldsymbol{X} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{99}]^\top$ over integer tokens, we center across integers and measure

$$E_k(\boldsymbol{X}) = \left\| \sum_{n=0}^{99} (\boldsymbol{x}_n - \bar{\boldsymbol{x}}) e^{-2\pi i k n/100} \right\|_2^2 . \tag{10}$$

At $k = 10$, ten cycles across the 100 ordered integers give a period of ten. Figure 5C reports the absolute-energy ratio $E_{10}(\boldsymbol{X}_{\text{intervention}})/E_{10}(\boldsymbol{X}_{\text{baseline}})$.

Model	Embedding E	After layer 1	Effective U
1-layer	0.510	0.556	1.025
8-layer	0.655	0.665	0.969

The selective reduction at the input and first hidden layer, with little change at the output, is informative: this intervention chiefly changes what integer tokens predict. Bare integer tokens, which it does not target, also change less at E and layer 1 (ratios $1.019/1.055$ in the one-layer model and $0.911/0.870$ in the eight-layer model). At deeper sites in the eight-layer model the period-ten ratios are 0.631, 0.741, and 0.605 after layers 2, 4, and 8, alongside broader spectral changes.

E.2 Bigram tokens and transfer across depth

This intervention asks whether shallow-model changes predict deeper-model changes beyond a selected numerical pattern. Merging `no longer`, for example, removes those uses from the remaining `longer` token: its continuation profile shifts away from `be` and toward `than`. The measured outcome is how pairs of surviving token vectors move relative to one another.

We jointly merge 200 selected GPT-2 token bigrams in a fixed 2B-token Pile stream, applying nonrecursive, left-to-right replacements within the original 512-token windows. The merges involve 274 original tokens and shorten the stream by 0.32%. Both training arms use the same expanded vocabulary (50,457 entries), paired initialization, original-text windows, order, and exposure-based learning-rate schedule. The one-layer attention-only and eight-layer attention-and-MLP models both have width 768 and 12 heads; within each depth, architecture and optimizer are fixed. Figure 5 uses seed 0 at both depths.

Frozen predictions and evaluation pairs. Before eight-layer training, we freeze the one-layer geometry changes and select 100 pairs per site using corpus-only forecasts: continuation-profile changes for $\boldsymbol{E}$ and a bigram-SVD output factor for $\boldsymbol{U}$. Each set contains 50 predicted separations and 50 approaches, requires at least 50,000 input and target occurrences per token before and after merging, and permits at most three appearances per token in each direction. The two sites therefore use different pair populations. We unit-normalize $\boldsymbol{E}$ without centering; for effective $\boldsymbol{U}$, we include the final RMSNorm gain, subtract the mean over the original 50,257 vocabulary rows, then unit-normalize. The measured change is the merged-minus-original difference in squared distances, used directly with no fitted scale or intercept. On these preselected pairs, one-layer changes correlate with eight-layer changes at $r = 0.69$ for E and $r = 0.70$ for effective U. Selection targets detectable corpus-predicted effects rather than the many nearly unchanged vocabulary pairs; shared endpoints mean that the 100 pairs are not 100 independent training replicates.

E.3 Control: continuation and target statistics act on different sites

The selective input-side effect above suggests a further question: do continuation and target statistics influence different interfaces? We test this using two paired seeds of four-layer, width-2048 models trained for 1B tokens, with attention restricted to a four-token context. Each baseline/intervention pair shares its architecture, optimizer, tokenizer, and token budget. Models use Llama-3.1 tokenization

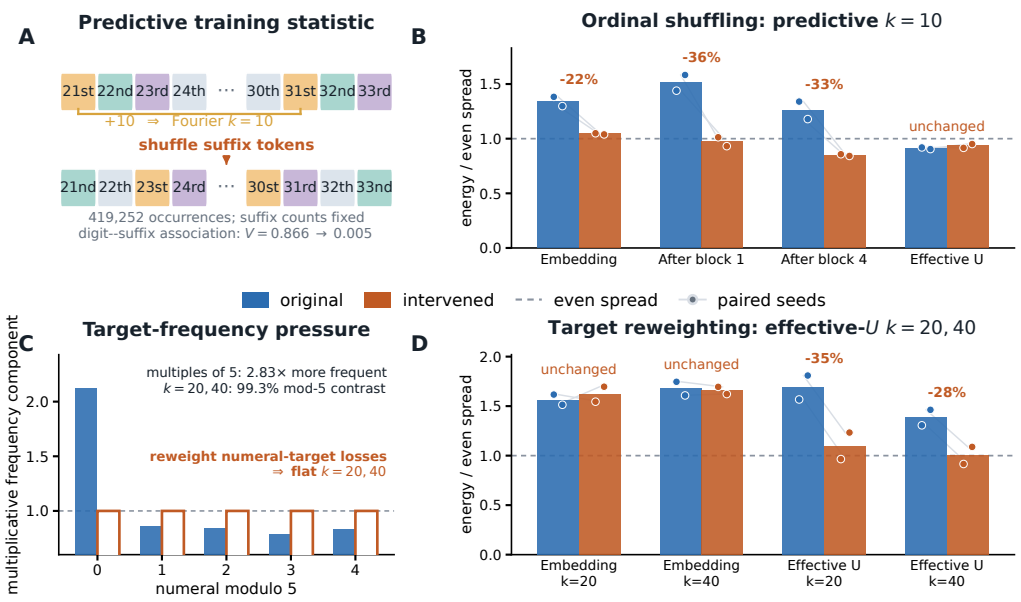


Figure 14: **Continuation and target statistics shape different components. A–B:** Ordinal shuffling reduces $k = 10$ in $\boldsymbol{E}$ and hidden states, largely preserving effective $\boldsymbol{U}$. **C–D:** Numeral-target reweighting reduces $k = 20, 40$ in effective $\boldsymbol{U}$, largely preserving $\boldsymbol{E}$. Matched four-layer, context-4 models train for 1B tokens; bars average two paired seeds, shown as dots. Energy is relative to an even spectrum, not a permutation null.

(128,256 vocabulary entries), 32 heads, and MLP width 7,168. Training packs 32 sequences of 512 tokens per batch, with learning rate 6×10^{-4}, 1% warmup, weight decay 0.05, gradient clipping at 1, and seeds 0 and 1. For this figure only, Fourier energy from Equation (10) is divided by $\frac{1}{50}\sum_{q=1}^{50} E_q$, so an even spectrum has value one. Bars are paired-seed means and dots retain the two seeds.

Continuation side: shuffling ordinal suffixes. We identify 419,252 numeral–ordinal-suffix occurrences and shuffle the suffix identities among these occurrences. This preserves stream length, every non-suffix token, the numeral tokens, and the marginal counts of `st`, `nd`, `rd`, and `th`, while breaking their relation to a numeral's final digit. Cramer's $\mathcal{V}$ between final digit and suffix falls from 0.866 to 0.00485. The normalized $k = 10$ component falls by 22% at $\boldsymbol{E}$, 36% after block 1, and 33% after block 4; effective $\boldsymbol{U}$ changes by +2%.

Target side: changing numeral-target training pressure. We compute centered log counts of numeral targets, remove their inverse-DFT $k = 20$ and $k = 40$ components, and convert the difference into a target-specific loss weight. The same weight is applied to every occurrence of a given numeral target; nonnumeral targets are unchanged. The input token stream and the empirical distribution of contexts conditional on each target therefore remain fixed, while the training objective no longer rewards the selected excess target-frequency modes. In the baseline counts, a decade-controlled multiples-of-five indicator explains $\boldsymbol{R}^2 = 0.993$ of the removed component and has frequency ratio 2.83. The normalized $k = 20$ and $k = 40$ components of effective $\boldsymbol{U}$ fall by 35% and 28%, while the corresponding embedding components remain largely unchanged.

The two interventions complement the tokenizer experiments: changing continuations primarily moves input geometry, whereas changing target pressure primarily moves output geometry. Shared structure at E and U can therefore arise from different aspects of the same prediction objective.