The Learning Dynamics of Geometric Structure in Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Language models trained on various tasks exhibit a rich *geometric structure* in the representation space, such as smooth manifolds for counting characters in a line. But how these geometric structures *emerge* during training, when they become *useful* for computation, and how their formation is reflected in the model *weights* remain poorly understood. In this work, we study these learning dynamics in a small transformer trained to predict line breaks in synthetic text at a fixed line width. Specifically, we find a *phase change* that is primarily driven by the emergence of a 1D ordinal manifold for character counting, coinciding with the onset of accurate line-break prediction. This geometric structure exerts a progressive causal influence on the task. Early in training, ablating the character count subspace has no effect, but as the manifold forms and converges towards its final geometry, the same ablation increasingly degrades loss and suppresses line-break token prediction. At the weights level, we show that this phase transition exhibits the signature of *saddle escape*: low-rank updates in the attention weights and a transient negative-curvature direction in the loss landscape. Finally, we sweep the weight initialization scale α, which monotonically slows the phase transition as α decreases. More broadly, our work highlights the geometric structure as a lens for understanding how transformers acquire task-relevant representations during training.

1 Introduction

Language models exhibit striking geometric structure in their internal representations. Periodic concepts like days of the week and times of year are arranged on circles in the residual stream [Kantamneni and Tegmark, 2025, Modell et al., 2025, Karkada et al., 2026]; ordinal quantities such as the number of characters in the current line live on 1D manifolds embedded in low-dimensional subspaces [Gurnee et al., 2025]; semantic and analogical relations are visible as linear directions in word-embedding geometry [Korchinski et al., 2025]. At convergence, these geometries are well-documented and increasingly well-explained [Karkada et al., 2026]. *But when and how do they emerge during training?*

Prior work has studied learning dynamics through the loss trajectory. Grokking and hidden progress [Liu et al., 2022, Barak et al., 2023], abrupt capability onsets such as induction-head formation [Olsson et al., 2022, Park et al., 2024], and the saddle-to-saddle dynamics of deep linear networks [Saxe et al., 2014, Jacot et al., 2022, Rhee et al., 2026] characterise *when* new capabilities appear and *why* they appear discretely. On the representation side, activations of trained transformers have been mapped as concrete geometric objects [Gurnee et al., 2025, Modell et al., 2025, Kantamneni and Tegmark, 2025]. In this work, we study the learning dynamics of these geometric structures: when a representational manifold first appears, when it begins to causally influence the downstream computation it supports, and how its emergence registers in the model's weights.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

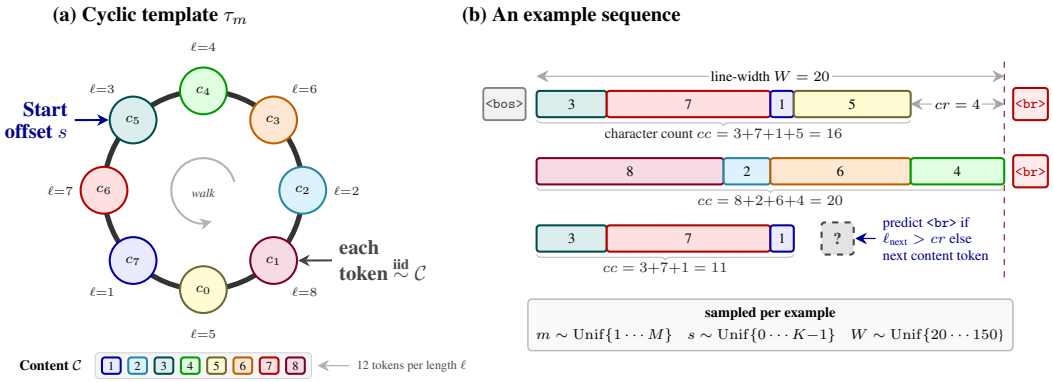

Figure 1: An overview of the **synthetic line-wrap dataset. (a)** The language is parameterized by one of $M = 32$ cyclic templates τ_m of length $K = 32$ (8 positions shown). Each template position contains a content token $c_i \in \mathcal{C}$; color denotes the token's character length $\ell(c_i) \in \{1, \ldots, 8\}$, with 12 content tokens per length class ($|\mathcal{C}| = 96$). **(b)** For each training example, we sample a template m, start offset s, and line width W, prepend <bos>, and then walk cyclically through τ_m from offset s. The character count cc accumulates the lengths of content tokens on the current line, with characters remaining $cr = W - cc$. At each position, the model must predict <br> iff the next content token would exceed the line width ($\ell_{\text{next}} > cr$); otherwise it predicts the next content token. After each <br>, cc resets to zero. The first row shows a break after an underfilled line, the second an exact-fill break, and the dashed gray box marks a non-
 prediction point.

We study this in a synthetic setting that isolates a single well-characterized representational geometry. Following Gurnee et al. [2025], we train a small transformer to predict line wrapping at a fixed line width — a task that requires maintaining a character count in the residual stream.

We summarize our *contributions* as follows.

- **The phase transition of the geometric structure.** We identify a sharp *phase transition* between training steps 400 and 1500 in which break-token F_1 rises from chance to $\sim$0.78 and the residual stream at layer 2 organizes into the 1D ordinal manifold for character count that Gurnee et al. [2025] document at convergence.
- **Causal effect of the representational geometry.** The character count subspace's causal role acquires its final direction during the transition itself — early-training steering is *anti-causal* (patching to a larger count *decreases* break preference) and inverts in lockstep with the formation of the manifold.
- **The phase transition in weight space.** The phase transition exhibits a *saddle escape*: QK weight updates collapse to low rank during the window and the loss Hessian opens a transient negative-curvature direction that recovers afterward.
- **Initialization scale controls the timing, not the outcome.** Across four orders of magnitude, the weight initialization scale α controls only the timing of the transition, not whether the capability is acquired.

2 Problem Setting

We study the task of linebreaking in fixed-width text. Given a text with a fixed line width W, the model must not only predict the next word, but also insert line breaks so that each line stays within the fixed width. Language Models solve this task by encoding the character count of the line on a 1D manifold in the residual stream [Gurnee et al., 2025], giving us a concrete geometric object whose emergence we can study and track. We train small transformers on a *synthetic* linebreak dataset (Figure 1) that preserves the core task structure while removing the semantic and lexical complexity of natural text.

Dataset. The vocabulary $\mathcal{V}$ contains four special tokens, {<pad>, <bos>, <eos>,
}, and 96 content tokens $\mathcal{C}$. Each content token $c_i \in \mathcal{C}$ has a character length $\ell(c_i) \in \{1, \ldots, 8\}$, with 12 tokens

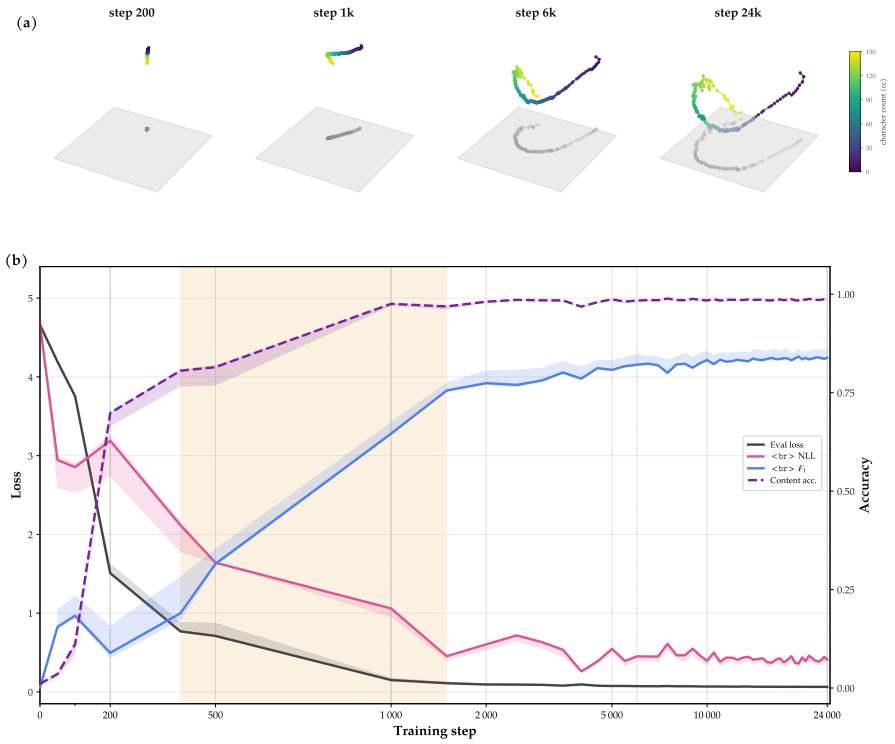

Figure 2: **Phase transition in the linebreak task. (a)** Class means $\{\mu_c^{(t,2)}\}$ at four training steps, projected onto the top 3 PCs of the final-checkpoint means (Eq. 1, $T = 24{,}000$). Color encodes character count $cc \in \{1, \ldots, 150\}$. **(b)** Eval loss $\bar{\mathcal{L}}^{(t)}$, `<br>` negative log likelihood $\bar{\mathcal{L}}^{(t)}_{\texttt{
}}$, `<br>`-token $F_1^{(t)}$, and content-token top-1 accuracy across training. Lines show the mean over the 5 seeds; shaded bands the min-max envelope. The amber band marks the transition window $t \in [400, 1500]$.

per length class. The synthetic dataset is defined by $M = 32$ cyclic templates $\tau_m \in \mathcal{C}^K$ of length $K = 32$, drawn once at construction time and held fixed across training (Figure 1a). To generate a sequence, we sample a template m, start offset s, and line width W, then walk cyclically through τ_m from offset s. We maintain a character count cc on the current line and remaining characters $cr = W - cc$. At each position, the model predicts the next content token if its length fits, $\ell_{\text{next}} \leq cr$, and is `<br>` otherwise; after a line break, cc resets to zero (Figure 1b). Thus the task decomposes into predicting the cyclic content stream and learning the linebreak rule. Each content token's character length is independently perturbed by ± 1 with probability 0.2, so the mapping from token identity to length becomes stochastic. Predicting `<br>` therefore requires tracking ℓ over the current line rather than just reading ℓ_{next} off the next token's embedding.

Line widths are drawn from $W \in [20, 150]$. Twelve evenly spaced widths $\mathcal{W}_{\text{out}} \subset [20, 150]$ are held out from training and validation sets and appear only at test time, separating in-distribution performance from generalization to widths the model has never seen. We sample $|\mathcal{D}_{\text{train}}| = 100$k training sequences (in-distribution widths only), $|\mathcal{D}_{\text{val}}| = 10$k validation sequences (in-distribution only), and $|\mathcal{D}_{\text{test}}| = 10$k test sequences mixing in-distribution and held-out widths. We use a fixed 2048 sequence subsample $\mathcal{D}_{\text{eval}} \subset \mathcal{D}_{\text{test}}$, drawn once for all subsequent analyses and reused across all training steps t.

Toy model. We train a small decoder-only transformer following the LLaMA 3 architecture, with two simplifications: standard multi-head attention (MHA) in place of grouped-query attention (GQA), and learned absolute positional embeddings (APEs) in place of RoPE. The embedding matrix $W_E \in \mathbb{R}^{|V| \times d}$ is weight-tied to the unembedding. The canonical configuration uses $L = 4$

layers, hidden dimension $d = 256$, $H = 16$ attention heads with $d_{\text{head}} = 16$, and MLP intermediate dimension $4d$. [1] See Appendix A.1 for full details and model hyperparameters.

3 Phase transition in geometric structure

We train the toy model from Section 2 for 24,000 steps across 5 random seeds on the line-wrap dataset and ask: *how does the geometry of the residual stream evolve as the model learns the linebreak task?*.

Task-progress metrics. At every checkpoint t, we evaluate on the fixed subsample $\mathcal{D}_{\text{eval}}$ of $N = 2048$ sequences, held identical across t so that variation in any reported metric reflects the model rather than the data. Let $\mathcal{B} = \{i : y_i = \texttt{
}\}$ denote the break token positions in $\mathcal{D}_{\text{eval}}$, $\mathcal{L}_i^{(t)} = -\log p_\theta^{(t)}(y_i \mid x_{<i})$ the per-position cross-entropy loss, and $\hat{y}_i^{(t)} = \arg\max_v p_\theta^{(t)}(v \mid x_{<i})$ the predicted next token. We track four task-progress metrics at each checkpoint: *(i)* the eval loss $\bar{\mathcal{L}}^{(t)} = \mathbb{E}_i\left[\mathcal{L}_i^{(t)}\right]$ averaged over all token positions in $\mathcal{D}_{\text{eval}}$; *(ii)* the
-NLL $\bar{\mathcal{L}}_{\texttt{
}}^{(t)} = \mathbb{E}_{i\in\mathcal{B}}\left[\mathcal{L}_i^{(t)}\right]$ averaged only over break positions; *(iii)* the
-token $F_1^{(t)}$ on $\mathcal{B}$; and *(iv)* content-token top-1 accuracy on $\{i : y_i \in \mathcal{C}\}$. The accuracies are computed from $\hat{y}^{(t)}$. We report the
-NLL alongside the average eval loss since break tokens are rare (roughly 1 in 20 positions) , so the average eval loss is dominated by content tokens and obscures the dynamics of the break decision.

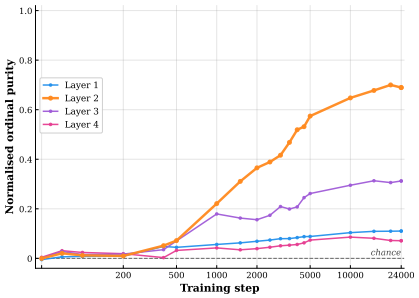

Figure 3: **Normalized ordinal purity** $\tilde{P}^{(t,l)}$ across training, by layer.

Character count manifold. At every checkpoint t, we extract the residual stream at layer l, $h_i^{(t,l)} \in \mathbb{R}^d$ at every token position in $\mathcal{D}_{\text{eval}}$ and group those positions by the character count of the current line, $cc \in \{1, \ldots, W_{\max}\}$ with $W_{\max} = 150$. Averaging within each group,

$$\mu_c^{(t,l)} = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} h_i^{(t,l)}, \qquad \mathcal{I}_c = \{i : cc_i = c\}, \tag{1}$$

collapses across-sequence variation (template, line width, position within the sequence) and leaves only the count-conditional component of the representation. We primarily analyze layer 2 for the manifold because the top three PCs of $\{\mu_c^{(T)}\}$ capture $\sim 95\%$ of variance at convergence (compared to $\sim 68 - 85\%$ for other layers). To visualize how the manifold forms during training (Fig. 2a), we fit a 3-component PCA on the mean-centered $\{\mu_c^{(T)}\}$ at the final checkpoint $T = 24000$ and project every earlier checkpoint $\{\mu_c^{(t)}\}$ onto this fixed basis. A shared basis ensures motion across panels reflects geometric change in the representation.

Ordinal purity. We project every residual-stream activation $h_i^{(t,l)}$ onto the top $k = 6$ PCs of the character count, cc manifold $\{\mu_c^{(t,l)}\}$, build a k nearest-neighbor graph in the projected space, and measure, for each point, what fraction of its k neighbors have a character count within $\tau = 5$ of its own:

$$P^{(t,l)} = 1 - \frac{1}{Nk} \sum_i \sum_{j \in \mathcal{N}_k(i)} \mathbf{1}\{|cc_i - cc_j| > \tau\}, \qquad \tilde{P}^{(t,l)} = \frac{P^{(t,l)} - P_0}{1 - P_0}, \tag{2}$$

where $N = |\mathcal{D}_{\text{eval}}|$, $\mathcal{N}_k(i)$ is the k-nearest-neighbor set of i in the projection, and $P_0 = P^{(0,l)}$ is the same statistic on the untrained model. The normalization puts the random-init chance level at $\tilde{P} = 0$ and a perfectly count-organized neighborhood at $\tilde{P} = 1$. A high $\tilde{P}$ means that activations close in the residual stream share nearly the same character count, so the count is encoded by *local* geometry —

[1] Token embeddings are mean-centered within each length class at initialization to remove a finite-sample bias in cc-conditioned averages at step 0. The trained model converges to the same solution; only step 0 geometry is affected.

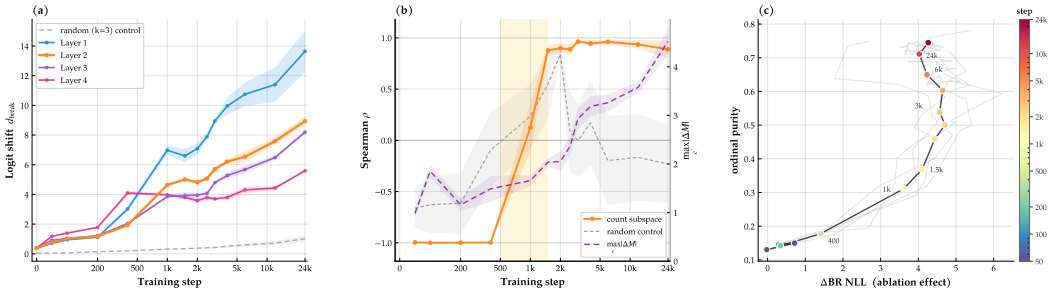

Figure 4: **Causal effect of the count subspace. (a)** Restricted-pair logit shift $d^{(t,l)}_{\text{break}}$ (Eq. 4) from ablating the rank-3 count subspace at each layer, against a random control averaged over five draws (grey dashed). **(b)** At $l = 2$: Spearman correlation $\rho^{(t,2)}$ between target count c' and $\Delta M^{(t,2)}(c')$ for the count-subspace patch (orange, left axis) and the random control (grey dashed); effect size $\max_{c'} |\Delta M^{(t,2)}(c')|$ (purple dashed, right axis). Amber band marks the phase-transition window $t \in [400, 1500]$. **(c)** Joint trajectory at $l = 2$ of ordinal purity $\tilde{P}^{(t,2)}$ (Eq. 2) against the <br> token NLL increase. Faint lines are individual seeds; bold trajectory is the mean across 5 seeds, colored by training step (colorbar).

the defining property of a 1D ordinal manifold. A low $\tilde{P}$ means neighbors are mixed across counts: whatever structure the projected space contains is not organized by count.

Results. The two subtasks resolve on markedly different timescales (Figure 2b). Content top-1 accuracy reaches $\sim 70\%$ by step 200 and saturates above 99% by step 1500 – once the active template and offset are identified, the next content token is deterministic. The break decision is harder.
-$F_1^{(t)}$ stays near chance through step 200, then undergoes a sharp *phase transition* through the shaded window $t \in [400, 1500]$, climbing to ~ 0.78; the
-NLL mirrors it, dropping from ~ 3 to ~ 0.5 nats over the same window.

This behavioral phase transition is accompanied by a geometric one in layer 2. The class means $\{\mu_c^{(t,2)}\}$ collapse to a point at step 200, form a short ordinal arc by step 1k, and develop into the hooked 1D curve at convergence (Fig. 2a). Ordinal purity quantifies the same trajectory (Fig. 3): $\tilde{P}^{(t,2)}$ leaves chance inside the transition window and continues climbing to ~ 0.70 by step 24k, while adjacent layers reach at most ~ 0.32 (layer 3) or stay below 0.12 (layer 1 & 4).

The behavioral and geometric phase transitions are coupled at onset and decoupled at offset:
-F_1 and $\tilde{P}^{(t,2)}$ leave chance in the same window, but after step 1500 F_1 gains less than 0.1 over the remaining training steps while $\tilde{P}^{(t,2)}$ more than doubles. Behavioral performance saturates long before the geometric structure does.

4 Causal effect of the geometric structure

Count-subspace ablation. We test when the count manifold becomes causally necessary for the break decision by removing it from the residual stream and measuring the resulting change in the model's
-vs-content token prediction. At each checkpoint t and layer l, let $V_k^{(t,l)} \in \mathbb{R}^{d\times k}$ be the matrix whose columns are the top $k = 3$ principal components of the mean-centered class means $\{\mu_c^{(t,l)}\}$ from Eq. 1; its columns form an orthonormal basis for the *count subspace* at (t, l). We ablate this subspace at every position by applying the orthogonal projection

$$\tilde{h}_i^{(t,l)} \;=\; h_i^{(t,l)} \;-\; V_k^{(t,l)} V_k^{(t,l)\top}\, h_i^{(t,l)}, \tag{3}$$

and resume the forward pass on the ablated stream to obtain logits $u'_i \in \mathbb{R}^{|\mathcal{V}|}$. Because <br> tokens occupy only $\sim 5\%$ of positions, the full cross-entropy and KL divergence between baseline and ablated distributions are dominated by content tokens and obscure the break decision. Following

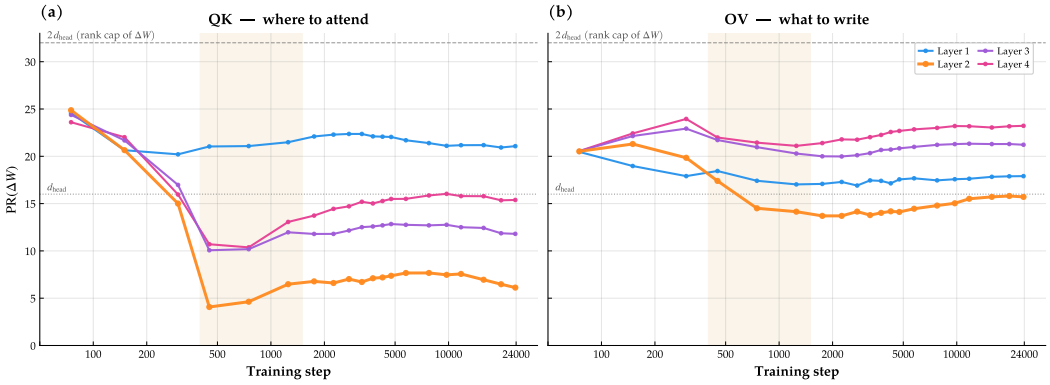

Figure 5: **Per-interval rank concentration of the attention weight updates.** Participation ratio $\overline{\mathrm{PR}}^{(t,l)}$ (Eq. 9) of the per-interval update ΔW at each layer, computed separately for the **(a)** QK and **(b)** OV circuits. Dashed line marks the rank cap $2\,d_{\text{head}} = 32$; dotted line marks the per-head rank $d_{\text{head}} = 16$. Amber band marks the phase-transition window $t \in [400, 1500]$.

Nielsen et al. [2026], we measure the intervention through the *restricted-pair logit shift*

$$d_{\text{break}}^{(t,l)} \;=\; \sqrt{\mathbb{E}_{i\in\mathcal{D}_{\text{eval}}}\left[\frac{1}{|\mathcal{C}|}\sum_{v\in\mathcal{C}}\left((u_{i,\texttt{
}} - u_{i,v}) - (u'_{i,\texttt{
}} - u'_{i,v})\right)^2\right]}\,, \tag{4}$$

the root-mean-square change in the
-vs-content logprobs across all eval positions and content tokens. By construction, d_{break} is invariant to additive shifts of the logit vector and so isolates the break decision from drift in the overall logit norm. We repeat the ablation with a random rank-k orthonormal basis, averaged over five draws (grey dashed curve in Fig. 4a) as a random control.

Character count patching. To probe for directional control, we steer the count component to a target count c' and measure the change in break preference. For a position with true count c_i at layer l, we replace its count-subspace component with that of a target class mean,

$$h_i^{(t,l)}(c') \;=\; h_i^{(t,l)} \;+\; V_k^{(t,l)} V_k^{(t,l)\top}\left(\mu_{c'}^{(t,l)} - \mu_{c_i}^{(t,l)}\right), \tag{5}$$

sweep c' over twelve evenly-spaced values in $\{1, \ldots, W_{\text{max}}\}$, and resume the forward pass. Let $M(h) = u_{\texttt{
}}(h) - \frac{1}{|\mathcal{C}|}\sum_{v\in\mathcal{C}} u_v(h)$ be the
-vs-content logit difference as a functional of the residual stream, and let

$$\Delta M^{(t,l)}(c') \;=\; \mathbb{E}_{i\in\mathcal{D}_{\text{eval}}}\left[M\big(h_i^{(t,l)}(c')\big) - M\big(h_i^{(t,l)}\big)\right] \tag{6}$$

be the average logit shift induced by patching to c'. We summarize the intervention with two complementary scalars. The Spearman rank correlation $\rho^{(t,l)} = \mathrm{Spearman}\big(c',\, \Delta M^{(t,l)}(c')\big)$ reports the *direction* of control — $\rho^{(t,l)} \to +1$ means patching to larger c' monotonically raises break preference. The effect size $\max_{c'} |\Delta M^{(t,l)}(c')|$ reports its *magnitude*. We compare against a random rank-k subspace as before.

Joint trajectory of the geometry & ablation effect. We trace the joint trajectory of the character count cc manifold and its causal effect at layer $l = 2$ across training in Figure 4c. We track ordinal purity $\tilde{P}^{(t,2)}$ (Eq. 2) paired with the
-token negative log likelihood increase

$$\Delta\mathrm{BR\,NLL}^{(t,l)} \;=\; \mathbb{E}_{i:\, y_i=\texttt{
}}\left[-\log p'(y_i \mid x_{<i}) + \log p(y_i \mid x_{<i})\right], \tag{7}$$

induced by the rank-3 count-subspace ablation of Eq. 3.

Results. Fig. 4 probes the causal influence of the representation geometry in three ways. **(i)** The character count manifold at $l = 2$ becomes causally necessary for the break decision precisely during

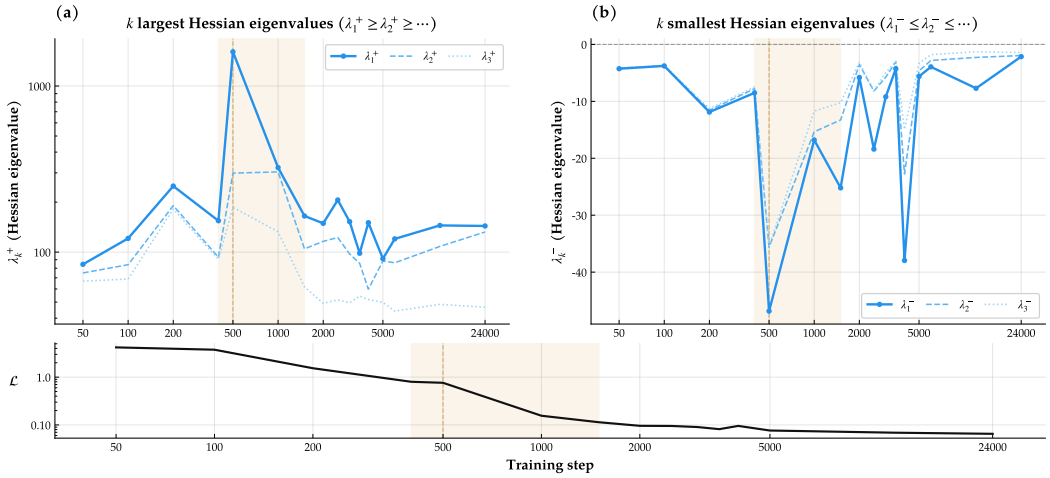

Figure 6: **Hessian eigenvalues across training.** Top: **(a)** three largest $\{\lambda_k^+(t)\}_{k=1,2,3}$ and **(b)** three smallest $\{\lambda_k^-(t)\}_{k=1,2,3}$ eigenvalues of the full-parameter loss Hessian $H_t = \nabla_\theta^2 \mathcal{L}_B(\theta_t)$, estimated by Lanczos iteration on a fixed validation batch B. Bottom: training loss $\mathcal{L}_B(\theta_t)$ on the same step axis. Amber band marks the phase-transition window $t \in [400, 1500]$; dashed vertical line marks the maximum-curvature checkpoint $t \approx 500$.

the phase-transition window $t \in [400, 1500]$: $d_{\text{break}}^{(t,2)}$ rises from 1.1 at step 100 to 5.2 at step 1500, exceeding the random rank-3 control by $\sim 5\times$, and continues to grow for the rest of the training. [2] **(ii)** Before the phase transition, the count subspace is *anti-causal*: $\rho^{(t,2)} = -1.0$ from step 50 through step ~ 400 — patching to a larger c' *decreases*
 token prediction. The manifold begins to form around step 1k (Fig. 2a), and $\rho^{(t,2)}$ rises in lockstep — flipping from -1 to $+1$ inside the transition window and stabilising by step 2k. The random control hovers near zero throughout. The geometric structure and its directional causal influence develops over the same phase transition window. **(iii)** The joint trajectory has two phases. Through step 1500, $\Delta\text{BR NLL}^{(t,2)}$ climbs to ~ 4 nats while $\tilde{P}^{(t,2)}$ improves from 0.13 to 0.31; after the phase transition, the causal effect saturates while ordinal purity continues climbing to ~ 0.75.

5 Phase transition in the weight space

We now ask what the linebreak phase transition looks like in weight space. The phase transition in Sections 3–4 is the moment a new computational capability (the break decision) and the geometric structure that implements it (the count manifold) come online together. Such discrete capability onsets are the canonical phenomenology of a *saddle escape* [Rhee et al., 2026]. Such an escape leaves two distinctive signatures: *(i)* the parameter update $\Delta\theta$ concentrates in a few singular directions and *(ii)* the Hessian of the loss transiently carries a negative-curvature direction that recovers once the escape completes. We track both across training and ask whether the phase-transition window $t \in [400, 1500]$ aligns with a saddle escape.

Attention weight updates. We test signature *(i)* on the attention weights, the parameters that route information across positions and are the natural candidates for a circuit-forming saddle escape. For each layer l and head h, we form the effective QK and OV circuits at checkpoint t,

$$W_{\text{QK}}^{(t,l,h)} = W_Q^{(t,l,h)\top} W_K^{(t,l,h)}, \qquad W_{\text{OV}}^{(t,l,h)} = W_O^{(t,l,h)} W_V^{(t,l,h)}, \tag{8}$$

and define the per-interval update between consecutive checkpoints, $\Delta W^{(t,l,h)} = W^{(t+1,l,h)} - W^{(t,l,h)}$, separately for QK and OV. We summarise the rank profile of each update via its singular-

[2] Layer 1 carries an even larger d_{break} which can be due to *cc* in superposition — a linear probe recovers *cc* at $l = 1$ with $R^2 \approx 0.4$, against $R^2 \approx 0.94$ at $l = 2$ — so the rank-3 subspace ablated at $l = 1$ entangles count with other features. The 1D manifold resolves only at $l = 2$, where we anchor the causal analysis.

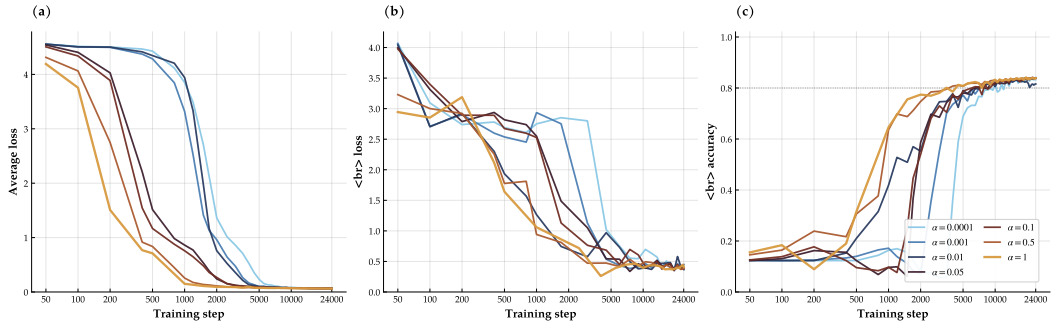

Figure 7: **Task-progress trajectories across initialization scales.** **(a)** eval loss $\bar{\mathcal{L}}^{(t)}$, **(b)**
-negative log likelihood $\bar{\mathcal{L}}^{(t)}_{\texttt{
}}$, and **(c)**
-$F_1^{(t)}$ across training. Color encodes α: warm tones for large α, cool tones for small α.

value participation ratio,

$$\mathrm{PR}\big(\Delta W^{(t,l,h)}\big) \;=\; \frac{\big(\sum_j \sigma_j\big)^2}{\sum_j \sigma_j^2}, \tag{9}$$

where $\{\sigma_j\}$ are the singular values of $\Delta W^{(t,l,h)}$. PR is a soft rank: $\mathrm{PR} = 1$ for an exactly rank-1 update, $\mathrm{PR} = r$ for a uniform rank-r spectrum. Each $W_{\mathrm{QK}}, W_{\mathrm{OV}} \in \mathbb{R}^{d\times d}$ has rank at most d_{head} (dashed reference line in Figure 5). Within each layer we aggregate across heads using Frobenius-norm weights,

$$\overline{\mathrm{PR}}^{(t,l)} \;=\; \frac{\sum_h \|\Delta W^{(t,l,h)}\|_F \cdot \mathrm{PR}\big(\Delta W^{(t,l,h)}\big)}{\sum_h \|\Delta W^{(t,l,h)}\|_F}, \tag{10}$$

so heads with negligible updates do not dominate the layer summary. A saddle escape predicts $\overline{\mathrm{PR}}^{(t,l)}$ to dip sharply inside the transition window $t \in [400, 1500]$ — the update concentrates into the unstable modes — and to recover toward its full-spectrum baseline once the escape completes.

Loss curvature. We test signature *(ii)* on the full-parameter loss Hessian. Let θ_t denote the parameters at checkpoint t and

$$\mathcal{L}_B(\theta) \;=\; \frac{1}{|B|} \sum_{(x,y)\in B} \mathrm{CE}(f_\theta(x),\, y) \tag{11}$$

the next-token cross-entropy loss on a fixed validation batch $B = 1024$ sequences. We form the loss Hessian $H_t = \nabla^2_\theta \mathcal{L}_B(\theta_t)$ and estimate its top and bottom three eigenvalues via Hessian-vector products and Lanczos iteration:

$$\lambda_1^+(t) \ge \lambda_2^+(t) \ge \lambda_3^+(t) \quad \text{from } H_t, \qquad \lambda_1^-(t) \le \lambda_2^-(t) \le \lambda_3^-(t) \quad \text{from } -H_t. \tag{12}$$

Wherever $\lambda_1^-(t) < 0$, the parameters lie at a saddle with a locally unstable descent direction. A saddle escape predicts $\lambda_1^-(t)$ to plunge below zero entering the transition window and recover to $\lambda_1^-(t) \approx 0$ once the escape completes; outside the window, all six eigenvalues should remain non-negative, locally convex along the unstable directions.

Results. **(i)** The QK participation ratio $\overline{\mathrm{PR}}^{(t,2)}_{\mathrm{QK}}$ collapses from ≈ 25 at step 50 to ≈ 4 at step 500 and recovers only to ≈ 6 post-transition (Fig. 5a) — the QK update at $l = 2$ concentrates into roughly four singular modes during the escape, against a rank cap of $2\,d_{\mathrm{head}} = 32$. Layers 3 and 4 show the same dip at smaller amplitude (minimum ≈ 10); Layer 1's QK updates stay near their starting rank throughout, consistent with the count being read off in superposition. The OV updates show no comparable dip (Figure 5b): all four layers drift smoothly toward $\overline{\mathrm{PR}} \approx d_{\mathrm{head}}$ over the full training run. The saddle escape concentrates in QK (where each head attends), not OV (what it writes). **(ii)** The smallest Hessian eigenvalue $\lambda_1^-(t)$ plunges from ≈ -5 at step 50 to ≈ -45 at step 500 — the deepest negative-curvature direction in the trajectory — and recovers toward zero by step 24k (Fig. 6b). The top eigenvalue $\lambda_1^+(t)$ spikes from ≈ 250 to ≈ 2000 at the same checkpoint (Fig. 6a),

a transient sharpening along the unstable direction. Smaller post-transition dips in λ_1^- around step ~ 4000 are consistent with finer secondary escapes, but the dominant event coincides exactly with the transition window. At the start of the phase transition, the QK weights updates at the manifold layer concentrate to near rank-4 and the loss Hessian opens a deep transient negative direction. The linebreak *phase transition* exhibits the dynamical fingerprint of a *saddle escape*.

6 Weight initialization scale & the phase transition

If the linebreak phase transition is a saddle escape, its timing should be controlled by the weight initialization scale: deep linear network theory predicts a logarithmic dependence, $\tau \sim \log(1/\alpha^2)$ [Rhee et al., 2026]. We sweep weight initialization scale α across four orders of magnitude in order to test this.

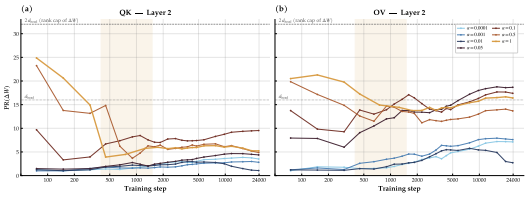

Figure 8: **QK and OV update participation ratio across initialization scales at $l = 2$.** $\overline{\mathrm{PR}}^{(t,2)}$ (Eq. 9) of the per-interval update ΔW at the character count manifold layer, computed separately for the **(a)** QK and **(b)** OV circuits. Dashed line marks the rank cap $2\,d_{\mathrm{head}} = 32$; dotted line marks the per-head rank $d_{\mathrm{head}} = 16$. Amber band marks the canonical-α phase-transition window $t \in [400, 1500]$.

Sweep. We train a family of otherwise identical models indexed by $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.05, 0.1, 0.5, 1\}$. Every linear and embedding matrix — including the tied embedding/unembedding — is initialized as

$$W_0 \sim \mathcal{N}\big(0,\, (\alpha\sigma_0)^2\big) \tag{13}$$

where σ_0 is the default initialization standard deviation. All variants share architecture, dataset, optimizer, learning-rate schedule, checkpoint schedule, and minibatch order — they are trained lockstep in a single process, three seeds per α — so only α changes across runs. We track the same task and weight-space metrics defined in Sections 3–5: the `<br>` negative log likelihood and `<br>`-F_1 trajectories (Figure 7) and the per-interval QK and OV participation ratios at $l = 2$ (Figure 8).

Results. The phase transition shifts monotonically later as α decreases. The transition window of the canonical $\alpha = 1$ run sits at $t \in [400, 1500]$; at $\alpha = 0.05$ it shifts to roughly $[1000, 2000]$, and at $\alpha = 10^{-3}$ to roughly $[1500, 2700]$. The QK participation ratio also changes shape with α. At large α ($\{0.5, 1\}$), $\overline{\mathrm{PR}}_{\mathrm{QK}}^{(t,2)}$ starts near 25 and dips sharply to ≈ 4 inside the transition window. As α shrinks, the starting value drops — ≈ 10 at $\alpha = 0.1$, ≈ 1 at $\alpha \leq 0.01$ — and the dip-and-recover collapses into a slow rise from below. Large α produces the canonical dip-and-recover signature of saddle escape. Small α initializes at the saddle itself, where ΔW_{QK} is concentrated from step 0 and the "escape" registers as a slow rise toward higher rank.

Across four orders of magnitude in α, every run reaches the same converged `<br>`-F_1; only the step at which it gets there changes. *α shifts the timing of the phase transition without changing whether the capability is acquired or not.*

7 Conclusion

We trace the formation of a representational manifold in a small transformer, end to end: the count subspace forms during a sharp phase transition, acquires its causal role over the same window, and leaves a saddle-escape fingerprint in the attention weights and the loss Hessian. Several open questions remain. First, how does this picture scale with model depth and width — do larger transformers show the same coupling between manifold formation and saddle escape, or does the increased representational capacity smooth the transition into a continuous process? Second, does the saddle escape correspond to the formation of a *circuit*, and if so, do other interpretable circuits form through analogous escapes? Third, how does this depend on task complexity — does a richer task induce a longer staircase of saddle escapes, each constructing a different representational object?

References

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit, 2023. URL https://arxiv.org/abs/2207.08799.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. When models manipulate manifolds: The geometry of a counting task. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/linebreaks/index.html.

Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity, 2022. URL https://arxiv.org/abs/2106.15933.

Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition, 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, 2026. URL https://arxiv.org/abs/2602.15029.

Daniel J. Korchinski, Dhruva Karkada, Yasaman Bahri, and Matthieu Wyart. On the emergence of linear analogies in word embeddings, 2025. URL https://arxiv.org/abs/2505.18651.

Ziming Liu, Ouail Kitouni, Niklas Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning, 2022. URL https://arxiv.org/abs/2205.10343.

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

Beatrix M. G. Nielsen, Emanuele Marconato, Luigi Gresele, Andrea Dittadi, and Simon Buchholz. Logit distance bounds representational similarity, 2026. URL https://arxiv.org/abs/2602.15438.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads, 2022. URL https://arxiv.org/abs/2209.11895.

Core Francisco Park, Maya Okawa, Andrew Lee, Hidenori Tanaka, and Ekdeep Singh Lubana. Emergence of hidden capabilities: Exploring learning dynamics in concept space, 2024. URL https://arxiv.org/abs/2406.19370.

Mark Rhee, Dhruva Karkada, and Jamie Simon. Deep linear networks are a surprisingly useful toy model of weight-space dynamics. *Learning Mechanics*, 2026. URL https://learningmechanics.pub/deep-linear-nets.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL https://arxiv.org/abs/1312.6120.

A Appendix

A.1 Toy model hyperparameters

The canonical d=256 configuration used throughout the main text is summarized in Table 1. All training runs use AdamW with a linear warmup followed by a cosine decay to η_{min}.

Architecture	
Model family	Llama-3 style decoder-only transformer
Attention	Multi-head (no GQA)
Positional embeddings	Learned absolute (max_seq_len $= 1100$)
Embedding/unembedding	Weight-tied
Activation	SiLU
Normalization	RMSNorm (pre-norm)
Layers L	4
Hidden dimension d	256
Heads H	16
Head dimension d_{head}	16
MLP dimension d_{mlp}	1024 ($= 4d$)
Vocabulary size $\lvert\mathcal{V}\rvert$	100 (96 content $+$ 4 special)
Total parameters	$\sim$3.5M
Initialization	
Linear, embedding	$\mathcal{N}(0,\, \sigma_0^2)$ with $\sigma_0 = 0.02$
Length-class centering	Yes (Appendix **??**)
Training	
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Learning rate η	3×10^{-4}
Min learning rate η_{min}	3×10^{-5}
LR schedule	Linear warmup $\rightarrow$ cosine decay
Warmup steps	200
Weight decay	0.01
Gradient clipping	1.0
Batch size	64
Total steps	24,000
Precision	bfloat16
Seeds (canonical experiments)	5
Seeds (initialization-scale sweep)	3 per α
Data	
Templates M	32
Template length K	32
Line width range	$W \in [20, 150]$
Length jitter probability	0.2
Train / val / test sequences	100k / 10k / 10k
$\lvert\mathcal{D}_{\text{eval}}\rvert$	2048

Table 1: Hyperparameters for the canonical d=256 model.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract states three claims that map directly to the paper's body sections: (i) a phase transition in `<br>`-token prediction coincides with the emergence of a 1D character-count manifold in the residual stream (Sec. 3); (ii) this manifold becomes progressively causally necessary for the break decision over the same window (Sec. 4); (iii) the transition exhibits saddle-escape signatures in attention weight updates and loss curvature (Sec. 5), with timing controlled monotonically by the initialization scale α (Sec. 6).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The conclusion discusses the main limitations of the work: results are demonstrated on a single synthetic line-wrap task and a small toy transformer (4 layers, $d = 256$), and the saddle-escape characterization is empirical rather than derived from a closed-form theory of the linebreak circuit.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [No]

Justification: The paper invokes the DLN prediction $\tau \sim \log(1/\alpha^2)$ to motivate the α-sweep but does not restate its assumptions or reproduce its proof, instead testing it empirically in Section 6.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 2 fully specifies the synthetic dataset (vocabulary, length classes, template construction, generation procedure, train/val/test splits) and the toy transformer architecture, and Sections 3–6 give the exact definitions of every metric, ablation, and sweep configuration.

Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [No]

Justification: Code and data are not released with the submission, but Section 2 specifies the synthetic dataset construction in full, the architecture follows the standard LLaMA recipe with the simplifications listed, and Sections 3–6 give exact definitions for every metric, ablation, and sweep — sufficient for an independent reimplementation. We plan to release code upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 2 specifies the dataset splits, vocabulary, and architecture; Section 3 gives the training horizon, seed count, and evaluation subsample; Section 6 details the α-sweep configuration.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: All trajectories in Sections 3–5 are reported as the mean over 5 random seeds with min–max envelopes capturing seed-to-seed variability (training initialization and minibatch order); the α-sweep in Section 6 uses 3 seeds per α trained lockstep so that only α varies across runs.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [No]

Justification: Compute resources (GPU type, memory, wall-clock time per run, and total project compute) are not reported in the current draft; the toy transformer is small enough that all experiments run on a single consumer GPU, and we will add a compute summary to the appendix in the camera-ready version.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research uses only synthetic data and a small toy transformer for mechanistic analysis, involves no human subjects or personally identifiable information, and conforms with the NeurIPS Code of Ethics in all respects.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The work is foundational mechanistic interpretability research on a synthetic toy task with a small transformer; it introduces no deployable model, dataset, or application with a direct path to societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper releases no pretrained models, scraped data, or other assets with misuse potential — only a small toy transformer trained on a synthetic line-wrap task.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The paper uses no external datasets or pretrained models; the LLaMA architecture and prior methods we build on (Gurnee et al. [2025], Rhee et al. [2026], Nielsen et al. [2026]) are properly cited, and standard open-source libraries used for training and analysis will be acknowledged in the camera-ready version.

Guidelines:

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

13. **New assets**

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

Answer: [No]

Justification: The paper does not release any new assets at submission time; the synthetic dataset construction and toy model architecture are fully specified in Section 2 for independent reimplementation.

Guidelines:

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

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

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

Answer: [No]

Justification: The paper involves no crowdsourcing or research with human subjects; all experiments use synthetic data generated from a fixed procedural construction.

Guidelines:

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

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

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

    Answer: [No]

    Justification: The paper involves no human subjects research; all experiments use synthetic data and a small toy transformer, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [No]

    Justification: LLMs were not used as a component of the core methodology.

    Guidelines:

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