DEEP LATENTS, LONG MEMORY: LATENT PERSISTENCE EXPLAINS TEMPORAL CORRELATIONS IN TRANSFORMER ACTIVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The correlations between elements of a time series stem from the structure of the process that generated it. Here, we apply this lens to the internal representations of Large Language Models (LLMs): We define the autocorrelation function of the neural activity as the average dot product between the activations of a fixed layer at two context positions separated by a time lag. First, we find that the autocorrelations of deeper layers develop a slowly-decaying tail over training. We conjecture that this decay reflects the abstraction of the encoded features: Deeper layers encode more abstract variables, an abstract variable governs a longer stretch of the input, and its code persists over that stretch. We then test this hypothesis in two settings. In LLMs trained on synthetic data with known hierarchical latent structure, the persistence of latent codes predicts the qualitative structure of autocorrelations, including their development during training, and a large fraction of the empirical measures. In open LLMs trained on real text, we take sparse-autoencoder features as proxies for latent variables, and define their persistence time as the number of contiguous tokens where the feature is active. We find that the persistence time linearly correlates with the magnitude of autocorrelations across datasets and LLMs. In particular, the persistence time increases with depth as autocorrelations do, implying that deeper latent features persist for longer context spans. We complement this analysis by explicitly ablating the persistent sparse-autoencoder features and observing a consequent decrease in the autocorrelations.

1 INTRODUCTION

Language is structured across many temporal scales: local constructions constrain nearby tokens, while abstract concepts such as topic and discourse states can remain relevant over much longer spans. Large Language Models (LLMs) adapt to this structure over training, developing representations of syntactic (Peters et al., 2018; Tenney et al., 2019; Manning et al., 2020; Diego Simon et al., 2024) and semantic features (Engels et al., 2024; Gurnee et al., 2025; Acevedo et al., 2026a). Understanding such representations, including their content and the mechanism of their formation, remains a central challenge for a theory of LLMs and deep learning at large (Michaud, 2026).

On the empirical side, a growing collection of methods probes the content of LLMs' representations, as well as their organisation and development through the layers. Sparse autoencoders linearly decompose activations into feature dictionaries (Huben et al., 2024), with recent temporal extensions separating slowly varying from rapidly changing information (Lubana et al., 2026; Bhalla et al., 2026; Luo et al., 2026). Jacobian lenses expose verbalizable concepts represented in the residual stream (Gurnee et al., 2026). Integration windows quantify the range of contextual information accumulated at each layer (Skrill & Norman-Haignere, 2023); While both local geometry (Hosseini & Fedorenko, 2023; King et al., 2026) and global intrinsic dimensionality (Cheng et al., 2025) measures point towards an increase of the abstraction of representations through the layers. Neighborhood comparisons quantify the similarity of abstract representations between points in semantic correspondence, such as translations (Acevedo et al., 2026b) and image-caption pairs (Huh et al., 2024). On the theoretical side, training LLMs on synthetic data with known latent structure (Cagnetta et al.,

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

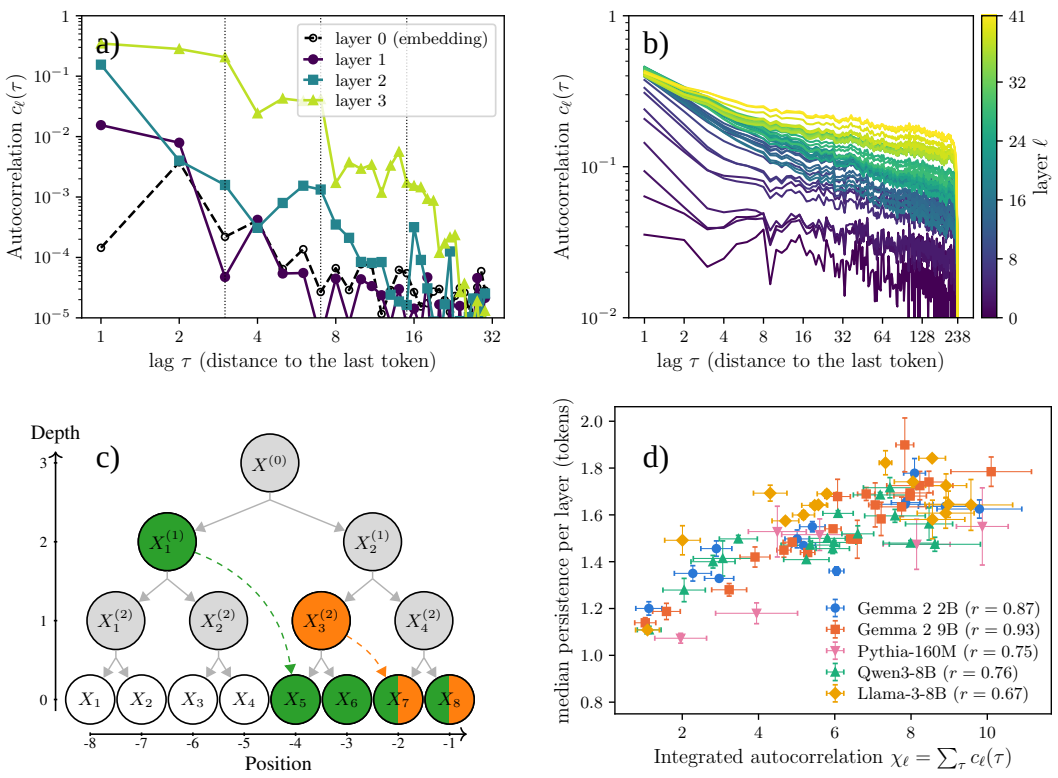

Figure 1: **Neural autocorrelations develop with depth due to persistently encoded latent features. a), b)** Autocorrelation $c_\ell(\tau)$ of the activations of each layer (Eq. 1) as a function of the lag τ to the last token, for a 5-block transformer trained on the Random Hierarchy Model (RHM; a) and for Gemma 2 9B (Gemma Team et al., 2024) (b, all 42 layers). Deep layers show a slower decay with the lag. **c)** In the RHM, the range of the autocorrelations reflects the predictive span of the latent variables. Because of the tree structure, a latent variable at height 1 above the leaves (orange) constrains the next 2 visible tokens; one at height 2 (green) constrains the next 4, one at height h the next 2^h. Latent variables that are relevant for predicting a longer span of the input remain encoded in the activations for longer. The dotted lines in a) mark the lags where each of said spans end. **d)** In natural language, we take sparse-autoencoder (SAE) features Huben et al. (2024) as proxies for latent variables. The median persistence of the SAE features of a layer, defined as the number of consecutive tokens over which a feature stays active, correlates linearly with the integrated autocorrelation $\chi_\ell = \sum_\tau c_\ell(\tau)$ of the same layer. One point per layer for five models; both quantities are averaged over The Pile and WikiText, giving the 1-std error bars shown, and r is the per-model Pearson correlation.

2024; Shai et al., 2024; Allen-Zhu & Li, 2023) allow for studying how representations are acquired, quantifying the sample cost of recovering abstract variables. In particular, these analyses show how deeper latent structures and longer-range statistical dependencies emerge progressively with data, training, and network depth (Cagnetta & Wyart, 2024; Garnier-Brun et al., 2025; Zhou et al., 2026; Schulz et al., 2026).

1.1 Our contribution

Bridging these two approaches requires translating insights obtained from synthetic datasets into observables that can also describe the behavior of pretrained language models. Here, we propose the autocorrelation function of neural activations across context positions as one such bridge. Given a set of LLM activations $\phi_t^{(\ell)}$ (layer ℓ, position $t = 1, \dots, T$) standardized so that $\mathbb{E}[\phi] = 0$, $\mathbb{E}[\|\phi\|^2] = 1$,

we define[1] an autocorrelation function of neural activations as

$$c_\ell(\tau) \coloneqq \mathbb{E}\left[\phi^{(\ell)}_{T-\tau} \cdot \phi^{(\ell)}_{T}\right]. \tag{1}$$

At the token level, if tokens are represented by one-hot vectors, the analogous correlation reduces, up to centering and normalization, to the probability of token coincidence. Eq. 1 extends this notion to continuous neural representations. Specifically, under the linear representation hypothesis (Park et al., 2024), a latent variable contributes to autocorrelations when it is encoded at both positions in aligned directions. Eq. 1 can be measured in synthetic data, where scope and identity of the underlying latent variables are known, as well as natural data, where SAE features offer an imperfect but tractable proxy for latent content.

Our main results are summarized graphically in Fig. 1: 1a) and b) show that autocorrelations grow with depth in both synthetic (Random Hierarchy Model, RHM) and natural language (see Appendix F for an extension to different models). We propose this is due to the latent hierarchical structure of language data (1c): if more abstract latents govern a longer stretch of text, and their code persists over that stretch, we expect them to produce a longer autocorrelation tail. The strong linear correlation between the integrated autocorrelation $\chi_\ell = \sum_\tau c_\ell(\tau)$ and the median persistence length (see Sec. 2.2) of SAE features, displayed in 1d), supports our hypothesis. Furthermore, removing the projection of persistent latent variables from the activations (RHM in Sec. 2.1 and natural language in Sec. 2.2.2) significantly decreases autocorrelations.

1.2 Further related work

Correlations at the token level. Two-point statistics of the tokens identify the underlying generative processes. In written texts, long-range correlations have been traced to the hierarchical structure of paragraphs and chapters (Alvarez-Lacalle et al., 2006). Power-law decay is indeed typical of hierarchical structures such as context-free grammars (Lin & Tegmark, 2017), whereas a Markovian structure would imply exponential decay. The same statistics govern learning: the decay of pairwise token correlations with the separation, combined with the decay of the next-token conditional entropy with the context length, predicts the exponents of data-limited neural scaling laws (Cagnetta et al., 2026). All these works characterise the input. Eq. 1 extends the same observable to representations, with the token-level function as its layer-0 limit.

Temporal structure of LLM representations. Several of the works cited above also track representations along context and layers, with different statistics. The curvature of Hosseini & Fedorenko (2023); King et al. (2026) can be thought of as a local autocorrelation, confined to time lag 1 and averaged across end positions. Our measure, instead, captures local as well as long-range structure. Skrill & Norman-Haignere (2023) use token perturbations to quantify effective integration windows, or receptive fields, finding a crossover from short-range, exponential behavior in the early layers to long-range, power-law behavior in the deeper layers. The increase of persistence time and integrated autocorrelation with depth is consistent with this observation. Gurnee et al. (2026) measure the autocorrelation of the dominant concept read out by a Jacobian lens, and find it largest in the intermediate layers. This observation supports the emergence of a global workspace, carrying abstract content that persists across the context. Our approach is complementary, as it focuses on raw activations, independently of how they affect the model's output.

Representations of models trained on data with controlled latent structure. Our analysis of synthetic data belongs to a line of work that trains networks on data with known latent variables and asks how these are represented. Allen-Zhu & Li (2023) train GPT-2 models on synthetic context-free grammars and show that the hidden states linearly encode the latent variables at the boundaries of their spans. Shai et al. (2024) show that transformers trained on hidden Markov models linearly represent, in the residual stream, the belief state over the hidden variables of the generator, reproducing its geometry within the probability simplex. In this language, the persistence of a code over the span of a latent variable is a belief about that variable that stays fixed until the span closes. On hierarchical data, attention maps and linear probes show that latent variables are resolved sequentially over training and depth (Garnier-Brun et al., 2025; Cagnetta et al., 2025; Zhou et al., 2026). These

[1] Eq. 1 is restricted to last-token correlations, using the T token as an "anchor", or reference token. We make this choice focusing on autoregressive LLMs.

works establish that latent variables are linearly readable from the activations. The regression of Eq. 2 is the analogue of their probes, which we first use to understand for how long and along which directions a latent variable remains encoded, and turn the answer into an observable that transfers to natural language.

2 RESULTS

2.1 AUTOCORRELATIONS WITH CONTROLLED HIERARCHICAL STRUCTURE

In this section, we look at a GPT-2-style model trained on RHM data, and describe how neural autocorrelations across layers reflect the latent hierarchical structure of the data.

Random Hierarchy Model. The RHM is a family of data-generating processes with controlled hierarchical structure, introduced in (Cagnetta et al., 2024) as a testbed for studying deep representation learning. A sentence of tokens is generated top-down along a binary tree of fixed depth (Fig. 1c): Starting from the root, every symbol is replaced by a pair of symbols of the next level according to fixed rules. This hierarchy of hidden variables constitutes the latent hierarchical structure of the data. Since the rules are fixed, the entropy of the distribution of generated sentences is known. Hence, we can track precisely the progress of any learning model trained on RHM data by comparing the cross-entropy loss to such entropy. Here, we consider a model with enough layers to represent the latent hierarchical structure of the RHM, and given just enough data, parameters and training time to reach the optimal cross-entropy loss (details in Appendix A). Specifically, we use vocabulary size $v = 32$, $m = 8$ rules per symbol, and depth 5 for both transformer and data-generating process. All sequences have length $T = 2^5$; The model reads a beginning-of-sequence token [BOS] followed by $X_1, \dots, X_{31}$ and predicts $X_1, \dots, X_{32}$.

Content of the trained network. Since the ground truth latent variables of every sentence are known, we can ask directly which of them each activation encodes. Let $\mathbf{Y}$ collect the one-hot encodings of all $2^6 - 1 = 63$ nodes of the tree, tokens included, and fit the activation at position t as a linear function of $\mathbf{Y}$ by least squares,

$$\phi_t^{(\ell)} = A_t^{(\ell)} \mathbf{Y} + \epsilon_t^{(\ell)}, \tag{2}$$

where the residual $\epsilon_t^{(\ell)}$ is uncorrelated with every node. $A_t^{(\ell)}$ is a D-by-$(v \times 63)$ matrix, assigning a different D-vector to each possible value (out of v) of any latent variable Y. This block of v D-vectors is the code of Y. Hence, Eq. 2 is the best approximation of the activation as a sum of single-variable codes, with the residual collecting non-additive contributions. We call the *strength* of the code of Y the share of an activation's variance occupied by the code. Appendix B details the method, its relationship with linear probes, and the complete encoding maps. Here we simply summarize the results:

- Layer 1 assembles depth-1 variables at even positions: the boundaries of the pairs generated by the same parent. Specifically, using the tree in Fig. 1c as reference: $X_1^{(2)}$ is encoded at position 2, $X_2^{(2)}$ at position 4, and so on. Odd positions, instead, encode the current token (X_1 at position 1, X_3 at position 3, and so on). These variables account for 90–100% of the activations' variance across positions.
- Layer 2 adds one level: depth-2 variables are encoded at multiples of 4, where the spans of tokens generated by the same grandparent end ($X_1^{(1)}$ at position 4). The other positions replay the roles of layer 1. Here persistence begins: depth-1 variables stay encoded, at decreased strength, until the span of their parent closes. The explained variance is 80–100%.
- Layer 3 continues the pattern: depth-3 variables are stored at 82–90% of the variance at the boundaries of the generated token span, while depth-2 variables persist until the span of the parent closes. What is new is, first, a partial encoding of depth-4 variables before their span is complete and, more generally, of the open ancestors of each position, suggesting a decoding operation rather than the encoding of the previous layers, accompanied by a significant non-additive component, reaching up to 40% of the variance at some positions. Secondly, not only depth-2, but also depth-3 variables become persistent at this layer, until the span of their parent closes.

- Finally, layer 4 reproduces layer 3, consistently with the fact that layer 3 has already reached depth-4 variables, while layer 5 turns to the output, i.e. the next-token probability (their autocorrelations are shown in Fig. 6 and discussed in Appendix C).

Composition of the autocorrelations. Substituting Eq. 2 into Eq. 1 splits the autocorrelation exactly into a part that is linear in the codes and a remainder,

$$c_\ell(\tau) = c_\ell^{\text{code}}(\tau) + c_\ell^{\text{res}}(\tau), \quad c_\ell^{\text{code}}(\tau) \propto \text{Tr}\left[A_T^{(\ell)}\,\text{Cov}(\mathbf{Y})\,A_{T-\tau}^{(\ell)\top}\right], \quad c_\ell^{\text{res}}(\tau) \propto \mathbb{E}\left[\epsilon_{T-\tau}^{(\ell)} \cdot \epsilon_T^{(\ell)}\right], \tag{3}$$

with the normalisation of Eq. 1. Expanding $\text{Cov}(\mathbf{Y})$ over pairs of nodes splits c_ℓ^{code} further into *same-node* terms, one variable encoded at both positions along aligned directions, which represents persistence, and *cross-node* terms, arising since all latent variables are correlated through the tree. Persistence alone predicts the shape of the curves in Fig. 1a, as explained by Fig. 2 for layer 3 (see Appendix D for additional details and the other layers). The reference activation is the last position, sitting above X_{31}, since X_{32} is never an input. Due to the tree structure, the variables that determine X_{32}'s probability sit on the tree diagonal: X_{31}, then $X_{15}^{(4)}$, parent of $X_{29:30}$ [2], then $X_7^{(3)}$, grandparent of $X_{25:28}$, and $X_3^{(2)}$, great-grandparent of $X_{17:24}$ (superscripts count levels from the root). The left panel compares the measured autocorrelation with the same-node terms of the three relevant latent variables (color code in the key). Each term is non-zero only over its window and vanishes where the target predates the completion of its variable: at lag 2 for $X_{15}^{(4)}$, lag 4 for $X_7^{(3)}$ and lag 8 for $X_3^{(2)}$. The right panel stacks the shares of the measured autocorrelation. The *open ancestors* of X_{31}, whose partial codes hold what the model has inferred about variables not yet determined by the visible tokens, can also be intended as a different representation of the main latent variables. With this caveat, the codes account for 60–95% of the measurement, the rest stemming mostly from the non-additive residual and confined to the first plateau.

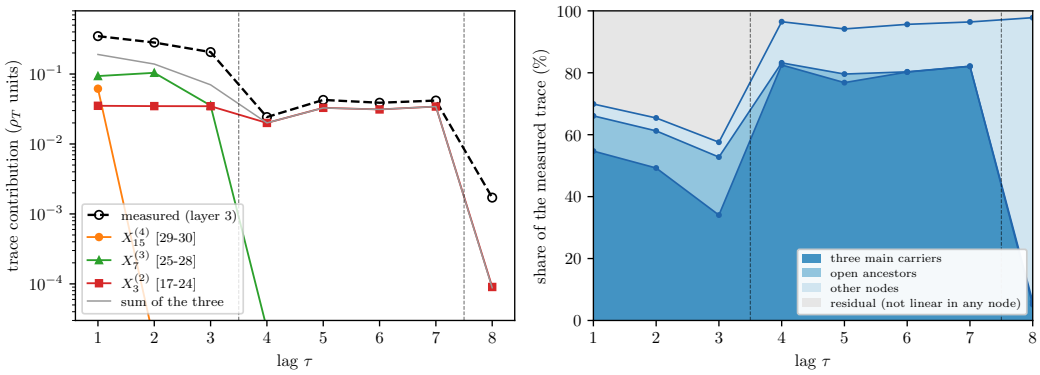

Figure 2: **RHM: composition of the autocorrelations of layer 3. Left:** measured autocorrelation $c_3(\tau)$ (dashed) against the same-node terms of Eq. 3 of the three relevant latents $X_{15}^{(4)}$, $X_7^{(3)}$ and $X_3^{(2)}$ (spans in brackets) and their sum (grey). Vertical dashed lines: the span ends of $X_7^{(3)}$ (between lags 3 and 4) and $X_3^{(2)}$ (between 7 and 8). **Right:** share of the measured autocorrelation accounted for, from bottom to top, by the three main carriers, by the open ancestors, by all other node codes including the cross-node terms, and by the non-additive residual c_3^{res}; the four bands sum to 100%.

Development over training. Persistence also predicts how the autocorrelation function should change during training. For deep transformers trained on RHM data, the learning curves proceed by stages, each stage corresponding to the resolution of one more level of the latent hierarchy (Cagnetta & Wyart, 2024; Cagnetta et al., 2025). Fig. 3a shows this for the validation loss of the last token, X_{32}, given the preceding ones, which decreases through a sequence of plateaus (the full autoregressive loss has identical stages; see Appendix E). The plateaus sit at the theoretical entropies of depth-d subtrees (dashed lines), compatibly with a model that has learned only a portion of the latent hierarchical structure, hence treats blocks of 2^d tokens as independent. Like the optimal loss, these

[2] $X_{i:j}$ represents the span of tokens $X_i, X_{i+1}, \ldots, X_j$.

plateau values are exact functions of the rules (Appendix A). We measure the autocorrelations at initialization and at one checkpoint on each plateau (vertical dotted lines). Fig. 3b shows the result for layer 3: the staircase is built outward, one plateau per stage. At initialization, the autocorrelation is flat, independently of the lag: attention is nearly uniform over the context, so that every position holds the average of its prefix (Appendix E). Training removes this uniform component, and the staircase grows one plateau at a time. At the checkpoint on stage 2, where depth-2 subtrees are learned, only lag 1 stands out (note the logarithmic scale). At stage 3, the plateau at lags 1–3 is in place; at stage 4, the plateau at lags 4–7 appears, then it settles at stage 5 where the curve reaches its final shape. The other layers follow the same stagewise behavior, with their own reach (Appendix E).

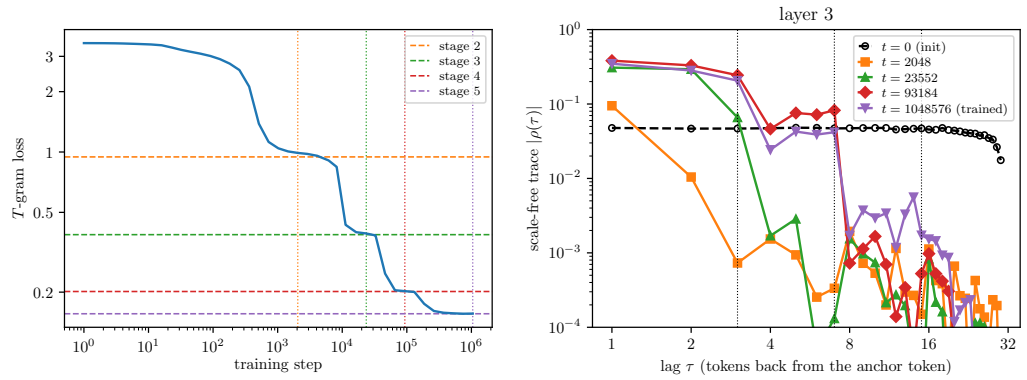

Figure 3: **RHM: the autocorrelations develop stage by stage. Left:** Validation loss of the last token, X_{32} given $X_1, \dots, X_{31}$, against the training step. Dashed lines: the plateau values of the stages, i.e. the conditional entropy of depth-d subtrees. Dotted lines: the checkpoints of panel b, coloured by stage. **Right:** Absolute value of the autocorrelation of layer 3 at initialisation (dashed) and at the four checkpoints; dotted vertical lines at lags 3, 7 and 15 as in Fig. 1a.

2.2 Natural language

In this section, we verify the relationship between autocorrelations, persistence, and depth of latent features in open LLMs trained on natural language.

Models and datasets. For natural language experiments, we used five LLMs for which open source trained SAE features are available: Gemma 2 2B and Gemma 2 9B (Gemma Team et al., 2024) with the residual-stream JumpReLU SAEs of Gemma Scope (Lieberum et al., 2024), Pythia-160M (Biderman et al., 2023) and Llama-3-8B (Grattafiori et al., 2024) with the top-k SAEs (Gao et al., 2025) released by EleutherAI (EleutherAI, 2024), and Qwen3-8B (Yang et al., 2025) with the top-k SAEs of Qwen-Scope (Deng et al., 2026). For datasets we used WikiText (Merity et al., 2017) and The Pile (Gao et al., 2020), standard open source corpora in English.

Integrated autocorrelation. By summing over time lags τ we get a scalar value quantifying the amount of autocorrelation of each layer, the *integrated* autocorrelation

$$\chi_\ell = \sum_\tau c_\ell(\tau). \tag{4}$$

χ is called *susceptibility* in equilibrium statistical mechanics, where integrated correlations are related to linear response to perturbations via the fluctuation–dissipation relations Goldenfeld (2018). For systems in which correlations change signs, Eq. 4 could average to zero even for finite values of $c_\ell(\tau)$, but this is not observed in our setup.

SAE and Persistence. SAE features are human-interpretable components extracted from the dense, entangled activation spaces of LLMs. SAEs do not represent the entire activation space: we report the fraction of the activation's variance explained by the SAE reconstructions in Appendix J.

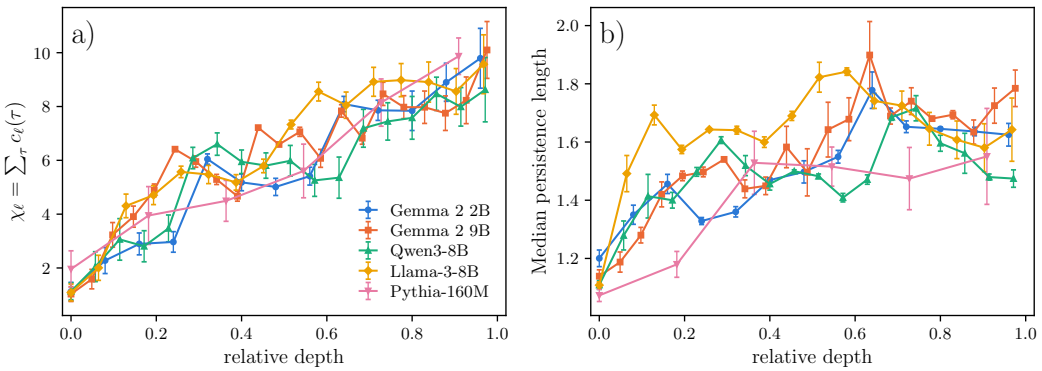

Figure 4: **Panel a)**: Layer dependence of the integrated autocorrelation for different models (see Eq. 4). **Panel b)**: Layer dependence of the median of the distribution of per-latent persistencies ψ (see Sec. 2.2). In both panels, error bars show one standard deviation computed from results on the Wikitext and The Pile datasets.

An SAE feature is *active* at a token position t when its code is non-zero. The persistence of a firing episode of an SAE feature is the number of consecutive tokens on which it is active: $\psi = e - s + 1$ for starting position s and ending position e (see Fig. 5, upper row). We report persistencies averaged per feature: record ψ for each firing episode of a given feature and average over firing episodes.

2.2.1 Layer dependencies

Fig. 4a and Fig. 4b show the layer dependence of the sum of autocorrelations (Eq. 4) and the median persistence time of SAE features, respectively, across models and datasets. Both quantities increase roughly monotonically with depth, although the median persistence tends to plateau or slightly decrease in the last quarter of the networks. The mean values of persistencies show a similar trend, which we report in Appendix H.

The long-lag autocorrelations shown in Fig. 1b contain a sequence-wide component shared across token positions. Appendix G separates this contribution by subtracting the mean representation of each sequence and recomputing the autocorrelations. The integrated autocorrelation $\tilde{\chi}_\ell$ of the remaining position-dependent fluctuations remains strongly correlated with SAE-feature persistence (see Table 7), and has non-monotonic layer dependence with an approximate plateau in the second half of the network (see Fig. 11). Interestingly, this layer profile is compatible with recent results in which measures of representational abstraction peak or saturate in intermediate layers Cheng et al. (2025); Acevedo et al. (2026b;a); Gurnee et al. (2026), but we leave further analyses regarding the decomposition of the autocorrelations for a future dedicated work.

2.2.2 Intervening on activations: ablation of SAE features

In this section, we verify that projecting out persistent features from the activations significantly decreases autocorrelations. Given the neural activity of an LLM layer ϕ_t [3] and an SAE feature i, ablation is performed by removing the projection of ϕ_t along the decoding vector $W_{\text{dec}}(i)$ of the SAE decoder:

$$\tilde{\phi}_t = \phi_t - \frac{(\phi_t, W_{\text{dec}}(i))}{||W_{\text{dec}}(i)||^2} W_{\text{dec}}(i). \tag{5}$$

We ablate token positions $t = s, s+1, ..., e-1$, such that the reference token e is not simultaneously ablated, then report the mean change in autocorrelations

$$\Delta = \frac{1}{e-s-1} \sum_{t=s}^{e-1} \mathbb{E}\left[(\tilde{\phi}_t - \phi_t) \cdot \phi_e\right] \tag{6}$$

[3] We omit the layer superscript in this section to ease notation, and restrict the analysis to the deepest layer for which there is an available SAE online.

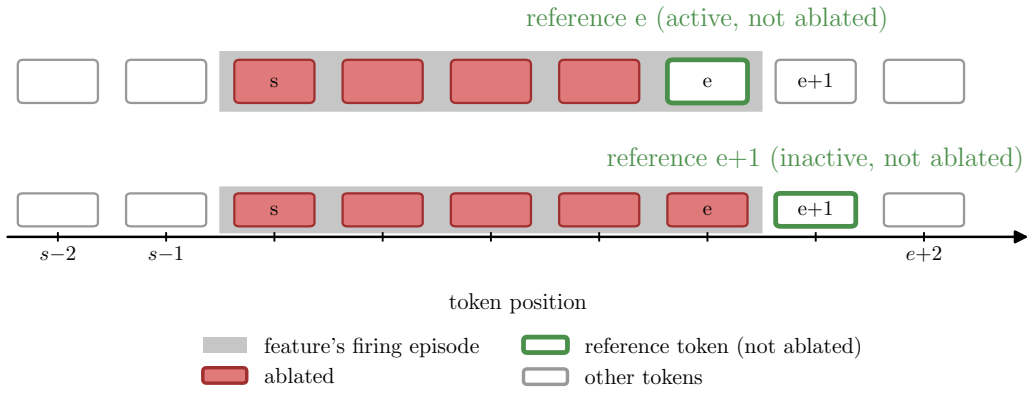

Figure 5: Scheme of the intervention experiment: In the gray areas, a given SAE-feature of interest is firing. s and e denote the starting and ending token positions of the firing episode. In the upper row, we measure the mean autocorrelations inside the feature's firing episode, taking as reference the last token of the firing episode, signaled with a green box. Then, we ablate SAE-features of the tokens colored in red using different protocols, and recompute the mean autocorrelation, to quantify how the ablation of SAE-features influences the autocorrelation. Results are reported in Table 1. In the bottom row, we show an example of a control experiment in which the reference token is outside the firing episode, for which the decrease in autocorrelations generated by ablating SAE-features is smaller, since the feature is not active in it.

Δ is further averaged over the 500 longest firing episodes of the layer's 400 most persistent latents, with at most 2 episodes per latent, and expressed as a percentage of the average clean value [4].

The third and fourth columns of Table 1 show the largest value of Δ obtained by removing one and three SAE features, as depicted in Fig. 5. The fifth column (a-control) shows the value of a control experiment in which we remove an SAE feature that was *active* at position s, but *not persistent* inside the window, for which we expect a much smaller Δ. The ablation of all active latents (sixth column) using the SAE intermediate representation does not remove all the baseline autocorrelations (i.e., $|\Delta| < 100$), as expected from imperfect SAE reconstructions (reported in Appendix J). The last three columns of Table 1 show the magnitude of Δ for the case in which the reference token is outside the measured firing episode of the latent ($e+1, e+2, e+3$, see Fig. 5 bottom row). As expected, the effect of the ablation is not negligible, but smaller than that measured using position e as reference (third column), and moreover it tends to decay with the distance to the e token, for all models except Pythia.

3 Discussion and perspectives

In this work, we explored the connection between the autocorrelations of neural activations, Eq. 1, and the encoding of latent variables. At the token level, with one-hot representations, Eq. 1 reduces to the centered probability of token coincidence. Under the linear representation hypothesis, extending to continuous representations measures the coincidence of latent codes, i.e., the persistence of variables encoded along aligned directions at both positions. Therefore, as at the token level, the functional form of autocorrelations reveals the structure of the underlying generating process. Note that autocorrelations are read from the raw activations, without downstream tasks or predefined features: the organization of the encoded variables is probed without even identifying their content. Thus, our results highlight the value of geometric observables for studying high-dimensional neural representations, taking a step toward connecting the geometry of LLM representations to the information they encode (Shai et al., 2024).

[4] Eq. 5 avoids overestimating the magnitude of a single feature ablation caused by the imperfect reconstruction of the SAE.

Model (Δ, %)	corpus	1 latent	3 latents	all latents	a-control	$e{+}1$	$e{+}2$	$e{+}3$
Gemma 2 2B	Pile	-3.9	-7.5	-58.8	-0.0	-0.4	$+0.0$	$+0.2$
Gemma 2 2B	WikiText	-3.2	-5.8	-68.3	$+0.0$	-0.5	-0.2	$+0.0$
Gemma 2 9B	Pile	-3.2	-6.2	-53.0	-0.0	-1.2	-1.3	-1.2
Gemma 2 9B	WikiText	-3.0	-4.6	-53.1	$+0.0$	-1.3	-1.0	-1.1
Pythia-160M	Pile	-12.2	-18.0	-43.7	-0.7	-8.4	-9.4	-7.0
Pythia-160M	WikiText	-13.3	-24.3	-60.4	-0.7	-8.1	-8.6	-9.0
Llama-3-8B	Pile	-4.4	-5.6	-68.1	-0.1	-2.1	-1.7	-1.5
Llama-3-8B	WikiText	-4.2	-4.4	-64.3	$+0.0$	-1.8	-1.4	-1.2
Qwen3-8B	Pile	-6.8	-20.3	-65.8	-0.2	-4.2	-4.6	-3.5
Qwen3-8B	WikiText	-7.3	-10.3	-67.6	-0.1	-3.8	-3.7	-3.5

Table 1: Change Δ (Eq. 6) of the mean integrated autocorrelations generated by the ablation of SAE-features, across models and two different corpora, and expressed as a percentage of the value of the baseline correlations in each model. See Fig. 5 for a scheme of the ablation setup, and Sec. 2.2.2 for the full discussion about the experiment. Columns: the persistent latent itself; it plus its two most persistent co-active latents; every active feature latent using the SAE's intermediate representation; an active but non-persistent latent (a-control); and the latent itself with the reference moved to $e{+}1$, $e{+}2$ and $e{+}3$ token positions, where the latent of interest was not found to be active.

Eq. 1 is invariant under a common change of orthonormal basis: a latent variable contributes only if it is encoded along aligned directions at both positions. While this property makes Eq. 1 a measure of persistence, nothing forces a model to write a variable along the same direction at different positions. Measures that include such an effect would rely on the full cross-covariance matrix, $\mathbb{E}\left[\phi^{(\ell)}_{T-\tau}\,\phi^{(\ell)\top}_{T}\right] \in \mathbb{R}^{D\times D}$. Comparing these measures with Eq. 1, which corresponds to the trace of the matrix, could establish how much of the persistence of a layer is carried by fixed directions, as in our reading of Eq. 1, and how much by codes that change orientation across positions.

Another direction is to connect correlations to the response to small perturbations. In equilibrium statistical mechanics, fluctuation–dissipation relations link the correlations of the unperturbed system to its linear response to weak external perturbations. Transformer representations do not constitute an equilibrium dynamical system, so such a relation is not expected to hold a priori. Nevertheless, first-order response matrices, such as the Jacobian lens Gurnee et al. (2026), have the potential to help us better understand how transformers process information internally. In this context, our results motivate asking whether the temporal structure of $c_\ell(\tau)$ predicts the lag-resolved response to small perturbations of earlier tokens or activations—for instance, whether layers with longer-ranged correlations also propagate perturbations over longer context ranges. Establishing such a fluctuation–response relation, or characterizing its violations, would connect the passive geometrical observable studied here to causal measures of information propagation.

AI USE STATEMENT

Generative AI tools were used as assistants for coding and for improving the clarity and presentation of the manuscript. They were not used to formulate the research questions or hypotheses, nor to design the experiments. All AI-assisted code was inspected and tested by the authors, and all AI-assisted text was reviewed and edited by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

To facilitate reproduction of our results, we will release all code used for training and analysis, together with the trained Random Hierarchy Model checkpoints, under the Apache License 2.0 after de-anonymization. The corresponding generated RHM datasets will be released under CC0 after de-anonymization. All natural-language experiments rely exclusively on publicly available pretrained language models, sparse autoencoders, and datasets, which are specified in the manuscript and remain subject to their respective licenses. We will provide the scripts necessary to reproduce the reported measurements, figures, and tables, together with the relevant hyperparameters and implementation details.

REFERENCES

Santiago Acevedo, Alessandro Laio, and Marco Baroni. Differential syntactic and semantic encoding in llms. *arXiv preprint arXiv:2601.04765*, 2026a. URL https://arxiv.org/abs/2601.04765.

Santiago Acevedo, Andrea Mascaretti, Riccardo Rende, Matéo Mahaut, Marco Baroni, and Alessandro Laio. A quantitative analysis of semantic information in deep representations of text and images. *Transactions on Machine Learning Research*, 2026b. URL https://openreview.net/forum?id=sBnaFSIuGR.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 1, learning hierarchical language structures. *arXiv preprint arXiv:2305.13673*, 2023.

Enrique Alvarez-Lacalle, Beate Dorow, Jean-Pierre Eckmann, and Elisha Moses. Hierarchical structures induce long-range dynamical correlations in written texts. *Proceedings of the National Academy of Sciences*, 103(21):7956–7961, 2006. doi: 10.1073/pnas.0510673103.

Usha Bhalla, Alex Oesterling, Claudio Mayrink Verdun, Hima Lakkaraju, and Flavio Calmon. Temporal sparse autoencoders: Leveraging the sequential nature of language for interpretability. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e4c2d7ec1f8c6020ee69c3101dcbaadc-Abstract-Conference.html.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 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. PMLR, 2023.

Francesco Cagnetta and Matthieu Wyart. Towards a theory of how the structure of language is acquired by deep neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=NaCXcUKihH.

Francesco Cagnetta, Leonardo Petrini, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How deep neural networks learn compositional data: The random hierarchy model. *Physical Review X*, 14:031001, 2024. doi: 10.1103/PhysRevX.14.031001. URL https://doi.org/10.1103/PhysRevX.14.031001.

Francesco Cagnetta, Alessandro Favero, Antonio Sclocchi, and Matthieu Wyart. Scaling laws and representation learning in simple hierarchical languages: Transformers versus convolutional architectures. *Physical Review E*, 112(6):065312, 2025. doi: 10.1103/qtd6-nl8p. URL https://doi.org/10.1103/qtd6-nl8p.

Francesco Cagnetta, Allan Raventós, Surya Ganguli, and Matthieu Wyart. Deriving neural scaling laws from the statistics of natural language. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.07488.

Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Lei Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0fD3iIBhlV.

Boyi Deng, Xu Wang, Yaoning Wang, Yu Wan, Yubo Ma, Baosong Yang, Haoran Wei, Jialong Tang, Huan Lin, Ruize Gao, Tianhao Li, Qian Cao, Xuancheng Ren, Xiaodong Deng, An Yang, Fei Huang, Dayiheng Liu, and Jingren Zhou. Qwen-scope: Turning sparse features into development tools for large language models. *arXiv preprint arXiv:2605.11887*, 2026.

Pablo J Diego Simon, Stéphane d'Ascoli, Emmanuel Chemla, Yair Lakretz, and Jean-Rémi King. A polar coordinate system represents syntax in large language models. *Advances in Neural Information Processing Systems*, 37:105375–105396, 2024.

EleutherAI. sparsify: Sparsify transformers with SAEs and transcoders. https://github.com/EleutherAI/sparsify, 2024. Checkpoints sae-pythia-160m-32k and sae-llama-3-8b-32x.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2024.

Ragnar Frisch and Frederick V. Waugh. Partial time regressions as compared with individual trends. *Econometrica*, 1(4):387–401, 1933.

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*, 2020.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *International Conference on Learning Representations*, 2025.

Jerome Garnier-Brun, Marc Mezard, Emanuele Moscato, and Luca Saglietti. How transformers learn structured data: Insights from hierarchical filtering. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AVXApuBCvN.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Nigel Goldenfeld. *Lectures on Phase Transitions and the Renormalization Group*. CRC Press, Boca Raton, FL, 2018.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, T. Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. Transformer Circuits Thread, July 2026. URL https://transformer-circuits.pub/2026/workspace/.

Eghbal A. Hosseini and Evelina Fedorenko. Large language models implicitly learn to straighten neural sentence trajectories to construct a predictive representation of natural language. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/88dddaf430b5bc38ab8228902bb61821-Abstract-Conference.html.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

Jack King, Evelina Fedorenko, and Eghbal A. Hosseini. Representational curvature modulates behavioral uncertainty in large language models. *arXiv preprint arXiv:2604.23985*, 2026. URL https://arxiv.org/abs/2604.23985.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, 2024.

Henry W. Lin and Max Tegmark. Critical behavior in physics and probabilistic formal languages. *Entropy*, 19(7):299, 2017. doi: 10.3390/e19070299.

Michael C. Lovell. Seasonal adjustment of economic time series and multiple regression analysis. *Journal of the American Statistical Association*, 58(304):993–1010, 1963.

Ekdeep Singh Lubana, Can Rager, Sai Sumedh R. Hindupur, Valérie Costa, Greta Tuckute, Oam Patel, Sonia Krishna Murthy, Thomas Fel, Daniel Wurgaft, Eric J. Bigelow, Johnny Lin, Demba Ba, Martin Wattenberg, Fernanda Viegas, Melanie Weber, and Aaron Mueller. Priors in time: Missing inductive biases for language model interpretability. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/45f7d161845e402f69c8f8b4d04fa447-Abstract-Conference.html.

Haoyan Luo, Mateo Espinosa Zarlenga, and Mateja Jamnik. Persistent sparse autoencoders: Learning feature timescales in language models, 2026. URL https://arxiv.org/abs/2607.17117.

C. D Manning, K. Clark, J. Hewitt, U. Khandelwal, and O. Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Eric J. Michaud. What is the purpose of interpretability?, 2026. URL https://ericjmichaud.com/interp-purpose.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 2024. URL https://proceedings.mlr.press/v235/park24c.html.

M. E. Peters, M. Neumann, L. Zettlemoyer, and W. Yih. Dissecting contextual word embeddings: Architecture and representation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1179. URL https://aclanthology.org/D18-1179.

Laura Ying Schulz, Daniel Mitropolsky, and Tomaso Poggio. Unraveling syntax: Language modeling and the substructure of grammars. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=sPIXjSkDFG.

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://arxiv.org/abs/2405.15943.

David Skrill and Samuel V. Norman-Haignere. Large language models transition from integrating across position-yoked, exponential windows to structure-yoked, power-law windows. In *Advances in Neural Information Processing Systems*, volume 36, pp. 638–654, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/020ad0ac6a1974e6748e4a5a48110a07-Abstract-Conference.html.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhiling Zhou, Tianhao Wang, and Zhuoran Yang. How transformers represent hierarchies: A local-to-global mechanism. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ScLLtpPoaE.

A Random Hierarchy Model: data, model and training

Generative process. The RHM (Cagnetta et al., 2024) generates sentences of $2^5 = 32$ tokens along a binary tree of depth 5, all of whose variables, latent and visible, take values in a set of v symbols. The root is drawn uniformly. At each level $l = 0, \dots, 4$ below the root, each of the v symbols is assigned m production rules, each replacing it with a pair of symbols of level $l + 1$; the vm right-hand sides are sampled uniformly without replacement among the v^2 possible pairs, so that no pair is produced by two different symbols (unambiguous rules). A variable is expanded by one of its m rules, chosen uniformly and independently of everything else. The k-th variable at level l is $X_k^{(l)}$; its depth $d = 5 - l$ is its distance from the tokens, and it spans the 2^d tokens $X_{(k-1)2^d+1}, \dots, X_{k2^d}$. The root ($d = 5$) spans the sentence and the depth-1 variables span pairs. We use $v = 32$ and $m = 8$: 63 tree nodes and $vm = 256$ grammatical pairs out of $v^2 = 1024$. Unambiguity makes the map from a sentence to its tree of latent variables deterministic, and we recover the latent variables of every sentence by parsing it bottom-up. The parse is verified by regenerating the sentence from the recovered tree.

Optimal loss. Since the generative model is a tree, the next-token posterior $\mathbb{P}(X_t \mid X_{<t})$ is computed exactly by belief propagation, and its conditional entropy $H_t := H(X_t \mid X_{<t})$, averaged over contexts, is the optimal cross-entropy loss at position t. H_t is a hierarchical sawtooth, close to $\log v$ at the positions opening a span and dipping at the positions closing one, the deeper the higher the closed variable. The sum over positions, 67.94 nats per sentence or 2.12 nats per token, is the optimal autoregressive loss. A model that has learned the tree only down to depth-d subtrees, i.e. that treats blocks of 2^d tokens as independent, has loss $(\log v + (2^d - 1)\log m)/2^d$ per token and loss H_{2^d} on the last token of the sentence; these values, for $d = 2, \dots, 5$, are the stages of Fig. 3.

Model. Decoder-only transformer in the GPT-2 layout: learned token and absolute positional embeddings of dimension $D = 768$, 5 pre-LayerNorm blocks with 12 attention heads and an MLP of width $4D$, a final LayerNorm and a linear readout; no dropout, no weight decay. Weights are initialized as standard Gaussians with the $1/\sqrt{\text{fan-in}}$ factors applied in the forward pass, and the readout is scaled by $1/D$, so that the initial predictions are close to uniform. The input is a beginning-of-sequence token followed by the first 31 tokens of a sentence, and the model is trained to predict $X_1, \dots, X_{32}$ from it with the cross-entropy loss. Layer $\ell = 1, \dots, 5$ of the main text is the residual stream after block ℓ, and layer 0 the sum of the token and positional embeddings.

Training. Adam with learning rate 3×10^{-3}, batch size 1024, linear warm-up over 8 steps and cosine decay to a tenth of the learning rate over 2^{19} steps, on a training set of 2^{28} sentences, for 2^{20} steps (four passes over the training set). The validation loss of the final checkpoint is within 0.033 nats per sentence of the optimum, and within 0.007 nats of the optimal H_t at every position. All results are for a single grammar and a single training run. The intermediate checkpoints used in Fig. 3 are discussed in Sec. 2.1; the initialisation checkpoint of the same run is the baseline for every measurement.

Measurements. Autocorrelations are computed from the activations of $P = 2^{25}$ sentences sampled from the grammar independently of the training set. The reference activation is the last one, above X_{31} (X_{32} is never an input); for each layer we accumulate the mean and the second moments of the activations above X_{31} and above $X_{31-\tau}$, from which the centered cross-covariance and the two self-covariances Σ_{31}, $\Sigma_{31-\tau}$ give $c_\ell(\tau)$ in closed form. For two independent centered activations the sampling variance of their average dot product is $\mathrm{Tr}(\Sigma_{31}\Sigma_{31-\tau})/P$, so the noise floor of $c_\ell(\tau)$ is $1/\sqrt{rP}$ with the effective rank $r = \mathrm{Tr}\Sigma_{31}\,\mathrm{Tr}\Sigma_{31-\tau}/\mathrm{Tr}(\Sigma_{31}\Sigma_{31-\tau})$; it lies below 10^{-4} at every lag and layer, and the curves are stable in P wherever they lie above it. The encoding regression of Eq. 2 is fitted on the moments of 2^{18} held-out sentences and checked against those of 2^{16} (Appendix B).

B Linear decomposition of the activations into latent codes

This appendix details the regression of Eq. 2, the reading of its coefficients, its relation to linear probes, and the encoding maps of the five layers that the item list of Sec. 2.1 summarises. Through-

out, $\phi := \phi_t^{(\ell)}$ is the activation of a fixed layer and position, $\Sigma_\phi := \mathrm{Cov}(\phi)$ its covariance, and every node X of the tree, latent or visible, is represented by its centered one-hot encoding $\mathbf{y}_X \in \mathbb{R}^v$, $(\mathbf{y}_X)_a = \delta_{X,a} - \mathbb{P}(X = a)$, with covariance Σ_X; $\mathbf{Y} \in \mathbb{R}^{63v}$ stacks the 63 encodings. All moments are accumulated over held-out sentences at $P = 2^{16}$ and 2^{18}, and the latent variables of every sentence come from the parse of Appendix A.

The joint regression. A linear function of the one-hot encoding of X is an arbitrary function of the symbol X, so the linear functions of $\mathbf{Y}$ are exactly the sums $\sum_X g_X(X)$ of single-node functions. The least-squares solution of Eq. 2, $A = \mathrm{Cov}(\phi, \mathbf{Y})\,\mathrm{Cov}(\mathbf{Y})^+$, is therefore the best approximation of the activation by such a sum. The residual ϵ, uncorrelated with every node by construction, collects what no sum of single-node functions captures: functions of several nodes at once, such as the rule chosen at a node, which is determined by the node together with its two children. Since the activation is a deterministic function of the visible tokens, and the tokens are among the nodes, ϕ is always a function of $\mathbf{Y}$, and the residual is its non-additive part. The decomposition is unique. The only linear dependencies among the encodings are the sum-to-zero constraints, one per node, because no node is a sum of single-node functions of the others: a parent is determined by the pair of its children, but not by a function of one child plus a function of the other. $\mathrm{Cov}(\mathbf{Y})$ has therefore rank exactly $63(v-1)$.

Partialling out. The coefficients of a joint regression obey the Frisch–Waugh–Lovell theorem (Frisch & Waugh, 1933; Lovell, 1963): the block A_X multiplying $\mathbf{y}_X$ equals the regression of ϕ on $\tilde{\mathbf{y}}_X$, the residual of $\mathbf{y}_X$ after its own regression on the encodings of all the other nodes. This property is what distinguishes Eq. 2 from a probe. Consider an activation that encodes only the token X_1, $\phi = g(X_1)$. X_1 and its parent $X_1^{(4)}$ are correlated through the rules, so ϕ is correlated with $\mathbf{y}_{X_1^{(4)}}$ and a linear readout of the parent from ϕ partially succeeds: with the rules used here, a function of one child linearly explains 11% of the variance of the parent, and the best sum of functions of the two children 23% (both values follow from the rule tables by belief propagation). We call this readability a *shadow*. In the joint regression, instead, $\phi = g(X_1)$ lies in the span of $\mathbf{y}_{X_1}$, one of the other nodes when the block of the parent is computed, so ϕ is orthogonal to $\tilde{\mathbf{y}}_{X_1^{(4)}}$ and $A_{X_1^{(4)}} = 0$ exactly. The argument applies to any node and any relative: the part of ϕ that is a sum of single-node functions of the other nodes contributes nothing to A_X, so $A_X \neq 0$ requires a component of the activation that correlates with the value of X beyond what any single other node implies. For a variable whose span is complete, this is a code computed from the conjunction of its children, as opposed to the shadow cast by the codes of the children. For a variable whose span is still open, or has not started, it is a prediction: a component correlated with the value the variable will actually take, which no function of a single visible node reproduces. The argument applies to any node and any relative: the part of ϕ that is a sum of single-node functions of the other nodes contributes nothing to A_X, so $A_X \neq 0$ requires a component of the activation that correlates with the value of X, beyond what any single other node implies.

Code strengths. The code of X is the block A_X of v vectors in $\mathbb{R}^D$, one per value of X, and its strength $S_X = \mathrm{Tr}(A_X \Sigma_X A_X^\top)/\mathrm{Tr}\,\Sigma_\phi$ is the variance of $A_X \mathbf{y}_X$ as a fraction of the activation variance. The explained fraction $\mathrm{Tr}(A\,\mathrm{Cov}(\mathbf{Y})A^\top)/\mathrm{Tr}\,\Sigma_\phi$ is the variance of the sum of all codes, and its complement is the residual. Every number in this appendix is a percentage of the activation variance.

Relation to linear probes. A linear probe for X is the regression in the opposite direction, from ϕ to $\mathbf{y}_X$. The least-squares solution is $W_X = C_X^\top \Sigma_\phi^+$, with $C_X := \mathbb{E}[\phi\,\mathbf{y}_X^\top]$. The quality of the probe is measured as the fraction of the variance of X that it recovers,

$$R_X^2 = \frac{\mathrm{Tr}\left(C_X^\top \Sigma_\phi^+ C_X\right)}{\mathrm{Tr}\,\Sigma_X}. \tag{7}$$

This is the infinite-data, unregularised limit of the probes trained by gradient descent in the works of Sec. 1.2. R_X^2 measures how well X can be read from the activations, and tree correlations inflate it: a parent, for instance, reads at 11–23$\%$ wherever its children are encoded. S_X measures how much of the activation is a code of X, and says nothing about readability: a code of a few percent

can be read perfectly if it occupies directions of its own, as happens for the carried depth-3 variable $X_1^{(2)}$ of layer 3, which reads at $R^2 = 0.7$–1 at positions 11–15, where its strength is 6–14%. Two further cautions apply to probes only. At positions 1 and 2 the grammatical prefixes number v and $mv = 256$, fewer than D, so any injective map, including a randomly initialised layer, makes every function of the prefix linearly readable; at later positions a random layer is a random-feature map of the prefix, from which a probe still recovers part of any variable. For these reasons we report strengths throughout and use R^2 only for cross-checks such as the one above.

Statistical controls. A node whose code is pure noise picks up a fraction $(v-1)/P$ of the residual variance, about 10^{-4} at $P = 2^{18}$. This spurious part scales as $1/P$ while a true code does not. Every strength we report is required to change by less than a factor 2 between $P = 2^{16}$ and 2^{18} (pure noise changes by 4): among the cells of Tables 2–5 above the display threshold of 0.05%, a single one fails the test and is blanked ($X_8^{(4)}$ at position 24 of layer 5, 0.06%). At initialisation the same regression explains 91–100% of the variance at every position with the codes of the visible tokens, the current one alone accounting for 70–100%, and assigns at most 1.1% to any latent variable.

Encoding maps. Tables 2–5 list the strengths of all codes at every position of the five layers (only positions 1–16 for layers 1 and 2, where the second half repeats the first). To keep the tables readable, the codes of the tokens and of the depth-1 and depth-2 variables, which come and go with the position, are listed by their role relative to the position (current token, last completed pair, and so on). The deeper variables, whose codes persist, are listed by name. The rest of this appendix walks through the layers, giving the numbers behind the item list of Sec. 2.1.

Layers 1 and 2 (Table 2). Layer 1 holds one code per position. Odd positions hold the current token: 100% at positions 1 modulo 4, and 87–94% at positions 3 modulo 4, where the variable of the pair just completed is retained at 3.5–7.4% and the previous tokens at up to 2.4%. Even positions hold the depth-1 variable of the pair they complete: 97.5–99.1% at positions 2 modulo 4, and 89–95% at the multiples of 4, where the current token takes 2–7%. The regression explains 98–100% of the variance, and nothing above the depth-1 level appears at the display threshold. Layer 2 adds the depth-2 variables at the multiples of 4: at 98–99% at positions 4, 12, 20 and 28, and at 70–79% at positions 8, 16 and 24, where the pairs of the 8-span just closed linger at 0.7–2% each (13 and 5% for the last two pairs at position 16), the previous depth-2 variable at 1.8%, and the depth-3 variable closing at positions 8 and 24 receives its first code, 2.5%. Positions 1 modulo 4 replay layer 1 (99–100% for the current token), while at positions 3 modulo 4 the depth-1 variable of the last completed pair is kept at 41–49% next to the current token at 29–43%, with a residual of 13–20%. This is the persistence of the item list: the code of a pair survives the position after the pair closes, and is dropped at the next one, when the span of its parent closes. The regression explains 80–100% of the variance.

Layer 3 (Table 3). The first eight positions extend the pattern by one level: the depth-2 variable completed at position 4 takes 99%, the depth-3 variable $X_1^{(2)}$ completed at position 8 takes 82.5% (and $X_3^{(2)}$, completed at position 24, 89.6%), with a residual of at most 5% at positions 1, 2, 4, 5, 8 and 9, and of 15–26% at positions 3, 6, 7, 10 and 11, where two codes coexist. Persistence now lasts for the whole span of the parent. The depth-2 variable $X_1^{(3)}$, completed at position 4, is retained at 70, 36 and 16% at positions 5–7 and disappears at position 8, when its parent closes ($X_5^{(3)}$: 69, 35, 13% at positions 21–23). The depth-3 variable $X_1^{(2)}$, completed at position 8, is retained at 71 and 42% at positions 9–10, at 6–14% at positions 11–15, and vanishes at position 16, when its parent $X_1^{(1)}$ closes; $X_3^{(2)}$ likewise decreases from 89.6% at position 24 to 2.6% at the last position, still within the span of its parent. The variables completed while a deeper code is carried get a smaller share: 30 and 26% for the depth-2 variables of positions 12 and 28, against 99 and 98% at positions 4 and 20, and 24–42% for the pairs of positions 10, 14, 26 and 30, against 97–100% at positions 2 and 18. From position 12 on, the activation also holds codes of variables whose span is still open: the depth-4 variable $X_1^{(1)}$ at 8, 4, 12 and 10% at positions 12–15, its open child $X_2^{(2)}$ at 2–4% and the open 4-span $X_4^{(3)}$ at 3–5%; in the second half, $X_2^{(1)}$ at 7, 3, 5 and 1% at positions 28–31, with $X_4^{(2)}$ and $X_8^{(3)}$ at up to 5%. These are the partial codes of the item list: by the partialling-out property, they are components of the activation correlated with the value of a variable that no visible

Layer 1, positions 1–16																
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	4
X_t	**100**	0.5	**91.5**	2.3	**100**	0.6	**91.6**	5.3	**100**	0.6	**93.3**	6.1	**100**	0.9	**86.9**	3.1
other X_i		0.2	1.4	1.2		0.2	1.5	1.0		0.2	1.3	1.1		0.2	2.4	1.2
$X_p^{(4)}$		**99.1**	5.1	**95.1**		**98.8**	4.9	**91.8**		**98.7**	3.8	**90.5**		**98.3**	7.4	**94.6**
sum of codes	100	99.8	98.0	98.7	100	99.6	98.0	98.2	100	99.5	98.4	97.7	100	99.5	96.8	99.0
explained	100	99.7	99.2	98.8	100	99.7	99.3	98.5	100	99.7	99.4	98.5	100	99.7	98.4	98.7
residual		0.3	0.8	1.2		0.3	0.7	1.5		0.3	0.6	1.5		0.3	1.6	1.3
Layer 2, positions 1–16																
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	4
X_t	**100**	0.1	**34.3**		**100**	0.1	**39.8**	0.2	**99.4**	0.1	**38.2**	0.1	**99.3**	0.2	**29.1**	0.5
X_{t+1}			0.8				0.8				0.9				0.9	
other X_i			0.2			0.1	0.2	0.1			0.2			0.1	0.4	0.3
$X_p^{(4)}$		**99.8**	**48.6**	0.4		**99.7**	**42.9**	1.1	0.1	**99.6**	**42.9**	0.3		**99.4**	**43.6**	12.7
$X_{p-1}^{(4)}$				0.3				0.7	0.1			0.2				5.4
$X_{p+1}^{(4)}$			0.8				0.8				0.9				1.2	
other $X_i^{(4)}$								3.3			0.1		0.3	0.1	0.1	
$X_q^{(3)}$				**99.1**				**78.5**	0.2			**98.7**				**70.3**
$X_{q-1}^{(3)}$								1.8					0.2		0.1	
$X_{q+1}^{(3)}$			0.7				0.7				0.7				1.9	
$X_1^{(2)}$ [1–8]								2.6								
sum of codes	100	99.9	85.5	99.8	100	99.9	85.3	88.3	99.8	99.8	84.0	99.4	99.8	99.8	77.4	89.3
explained	100	99.9	86.7	99.4	100	99.9	86.7	89.0	99.9	99.9	85.4	99.5	99.9	99.9	80.5	87.8
residual		0.1	13.3	0.6		0.1	13.3	11.0	0.1	0.1	14.6	0.5	0.1	0.1	19.5	12.2

Table 2: **Encoding maps of layers 1 and 2**, positions 1–16 (positions 17–31 repeat the pattern; the ranges quoted in Appendix B cover all 31 positions). Cells: percentage of the activation variance occupied by the code of each node at position t (strength S_X of Appendix B, $P = 2^{18}$), blank below 0.05%, bold at or above 20%. The row *closes depth* gives the depth of the deepest variable whose span ends at t. Tokens and depth-1 and depth-2 variables are listed by their role relative to t, with $p = \lfloor t/2 \rfloor$ and $q = \lfloor t/4 \rfloor$ the numbers of completed pairs and 4-spans: X_t is the current token and X_{t+1} the next, unseen one; $X_p^{(4)}$ is the last completed pair, $X_{p-1}^{(4)}$ the one before it and $X_{p+1}^{(4)}$ the pair containing X_t (open, odd t) or the following one (unseen, even t); likewise for the 4-spans $X_q^{(3)}$, $X_{q-1}^{(3)}$, $X_{q+1}^{(3)}$; the *other* rows sum the remaining nodes of that depth. Deeper variables are listed by name, with their span in brackets. *Sum of codes* adds the strengths of all 63 nodes, *explained* is the variance of the sum of the codes, and *residual* its complement, the part of the activation that is not a sum of single-node functions; the difference between the first two is the signed cross term of correlated codes.

Layer 3, positions 1–16

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	4
X_t	**100**	0.1	**33.7**		**27.0**		**28.4**	0.1	**26.6**		**32.3**		19.9		11.4	0.2
X_{t+1}			1.5				1.0				1.3				0.3	
other X_i			0.2			0.1	0.2	0.1		0.1	0.2	0.2	0.2	0.2	0.3	0.4
$X_p^{(4)}$		**99.8**	**41.1**	0.3		**39.5**	**24.9**	0.3		**41.9**	**30.2**	0.1		**23.6**	12.5	6.3
$X_{p-1}^{(4)}$				0.2				0.2				0.1				3.9
$X_{p+1}^{(4)}$			1.1			0.7	1.1			0.1	0.9			0.2	0.6	
other $X_i^{(4)}$			0.1			0.1	0.1	0.6			0.1	0.1	0.1	0.1	0.1	0.1
$X_q^{(3)}$				**99.2**	**70.2**	**36.0**	16.3	9.9	0.1			**30.0**	19.8	11.0	9.9	**33.7**
$X_{q-1}^{(3)}$								0.8	0.1						0.1	**24.9**
$X_{q+1}^{(3)}$			0.8			1.3	0.9			0.1	0.7	2.9	5.3	3.1	2.7	
other $X_i^{(3)}$										0.1	0.1	0.1	0.1	0.1	0.1	
$X_1^{(2)}$ [1–8]						1.0	0.3	**82.5**	**71.4**	**42.0**	6.5	14.1	8.5	7.1	9.4	
$X_2^{(2)}$ [9–16]										0.1		2.2	1.7	4.0	3.4	10.9
$X_1^{(1)}$ [1–16]										0.1		8.1	3.7	12.4	10.3	
sum of codes	100	99.9	78.5	99.7	97.3	78.7	73.2	94.4	98.1	84.6	72.3	58.1	59.3	62.0	61.1	80.4
explained	100	99.9	80.0	99.5	97.5	79.5	74.4	95.0	98.2	84.9	73.7	59.9	61.7	61.9	62.1	72.5
residual		0.1	20.0	0.5	2.5	20.5	25.6	5.0	1.8	15.1	26.3	40.1	38.3	38.1	37.9	27.5

Layer 3, positions 17–31

t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	
X_t	**92.5**	0.1	**37.1**		**27.3**		**32.1**		17.5		**39.9**		**25.2**	0.1	**30.4**	
X_{t+1}			1.6				1.1				1.3				1.0	
other X_i			0.2			0.1	0.2			0.1	0.2	0.2	0.2	0.2	0.2	
$X_p^{(4)}$	1.1	**97.3**	**36.9**	0.2		**34.8**	**25.9**	0.1		**33.6**	**28.6**	0.1		**30.1**	**28.3**	
$X_{p-1}^{(4)}$	0.4	0.4	0.1	0.2				0.1				0.1				
$X_{p+1}^{(4)}$			1.1			0.8	1.1			0.1	1.0			0.2	0.9	
other $X_i^{(4)}$		0.2	0.1	0.3	0.3	0.3	0.1	0.7	0.5	0.4	0.1	0.4	0.3	0.3	0.1	
$X_q^{(3)}$	4.4	1.5	0.2	**97.7**	**69.0**	**35.3**	12.8	4.6				**26.0**	**20.0**	14.4	7.4	
$X_{q-1}^{(3)}$				1.0	0.8	0.7	0.1	0.4								
$X_{q+1}^{(3)}$			0.8			1.5	0.9			0.1	0.8	3.9	4.6	4.3	1.3	
other $X_i^{(3)}$								1.1	1.4	1.2	0.2	1.0	0.6	0.8	0.1	
$X_3^{(2)}$ [17–24]						1.2	0.2	**89.6**	**78.6**	**44.0**	3.5	12.5	7.7	6.6	2.6	
$X_4^{(2)}$ [25–32]										0.1		2.2	1.4	1.6	0.4	
$X_2^{(1)}$ [17–32]										0.1		7.0	2.9	4.5	0.7	
sum of codes	98.4	99.4	78.0	99.4	97.4	74.9	74.5	96.7	98.1	79.9	75.5	53.6	63.1	63.2	73.3	
explained	97.3	99.0	79.4	99.0	97.4	75.5	76.0	96.9	97.8	79.8	77.0	54.5	65.1	63.9	75.3	
residual	2.7	1.0	20.6	1.0	2.6	24.5	24.0	3.1	2.2	20.2	23.0	45.5	34.9	36.1	24.7	

Table 3: **Encoding map of layer 3**, all positions. Cells: percentage of the activation variance occupied by the code of each node at position t (strength S_X of Appendix B, $P = 2^{18}$), blank below 0.05%, bold at or above 20%. The row *closes depth* gives the depth of the deepest variable whose span ends at t. Tokens and depth-1 and depth-2 variables are listed by their role relative to t, with $p = \lfloor t/2 \rfloor$ and $q = \lfloor t/4 \rfloor$ the numbers of completed pairs and 4-spans: X_t is the current token and X_{t+1} the next, unseen one; $X_p^{(4)}$ is the last completed pair, $X_{p-1}^{(4)}$ the one before it and $X_{p+1}^{(4)}$ the pair containing X_t (open, odd t) or the following one (unseen, even t); likewise for the 4-spans $X_q^{(3)}$, $X_{q-1}^{(3)}$, $X_{q+1}^{(3)}$; the *other* rows sum the remaining nodes of that depth. Deeper variables are listed by name, with their span in brackets. *Sum of codes* adds the strengths of all 63 nodes, *explained* is the variance of the sum of the codes, and *residual* its complement, the part of the activation that is not a sum of single-node functions; the difference between the first two is the signed cross term of correlated codes.

Layer 4, positions 1–16																
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	4
X_t	**100**		**34.4**		**53.3**		**32.5**	0.1	**66.4**		**35.1**		**43.1**		**25.5**	0.2
X_{t+1}			5.7		0.2		4.7				5.0		1.0		1.6	
other X_i			0.2		0.1	0.2	0.2	0.1		0.2	0.2	0.2	0.2	0.2	0.3	0.4
$X_p^{(4)}$		**99.9**	**20.2**	0.2		**29.2**	12.9	0.3		**49.9**	15.0	0.1		**22.2**	8.2	5.4
$X_{p-1}^{(4)}$				0.2				0.2								3.2
$X_{p+1}^{(4)}$			1.5		0.1	8.7	1.6			0.8	1.4		0.3	13.9	0.9	
other $X_i^{(4)}$			0.1		0.1	0.1	0.1	0.6		0.1	0.1	0.1	0.2	0.1	0.1	0.1
$X_q^{(3)}$				**99.3**	**28.8**	13.9	7.9	9.6				**24.1**	6.2	4.5	5.5	**29.1**
$X_{q-1}^{(3)}$								0.8								**21.8**
$X_{q+1}^{(3)}$			1.0		0.1	2.0	0.9			0.3	0.8	11.0	4.2	4.8	2.2	
other $X_i^{(3)}$						0.1				0.2	0.1	0.1		0.1	0.1	
$X_1^{(2)}$ [1–8]					0.1	1.2	0.2	**82.9**	**26.0**	15.1	3.0	8.7	2.4	2.4	4.9	0.3
$X_2^{(2)}$ [9–16]										0.2		2.0	0.7	1.7	2.0	9.7
$X_1^{(1)}$ [1–16]										0.1		4.2	0.9	3.6	5.0	7.5
sum of codes	100	100	63.2	99.7	82.8	55.4	61.2	94.5	92.6	66.8	60.8	50.7	59.4	53.7	56.6	77.8
explained	100	100	66.7	99.5	83.7	58.6	64.0	95.0	92.7	68.3	64.0	55.2	61.4	57.1	58.9	71.7
residual			33.3	0.5	16.3	41.4	36.0	5.0	7.3	31.7	36.0	44.8	38.6	42.9	41.1	28.3

Layer 4, positions 17–31															
t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–
X_t	**87.4**		**36.9**		**53.4**		**35.8**		**48.6**		**42.6**		**47.7**		**36.4**
X_{t+1}			5.7		0.2		4.6				4.1		1.0		3.8
other X_i		0.1	0.2	0.2	0.1	0.2	0.2	0.2	0.1	0.2	0.2	0.3	0.2	0.2	0.2
$X_p^{(4)}$	0.2	**84.8**	17.4	0.1		**24.8**	14.3	0.1		**30.5**	15.7			**24.2**	16.8
$X_{p-1}^{(4)}$	0.1	0.1		0.1				0.1							
$X_{p+1}^{(4)}$			1.6		0.1	9.8	1.6			2.3	1.4		0.3	17.8	1.4
other $X_i^{(4)}$		0.1	0.1	0.2	0.2	0.2	0.1	0.5	0.2	0.3	0.1	0.2	0.3	0.2	0.1
$X_q^{(3)}$	0.6	0.4	0.1	**62.4**	**23.7**	11.4	6.5	3.0				13.3	5.8	4.6	3.9
$X_{q-1}^{(3)}$				0.4	0.2	0.2		0.3							
$X_{q+1}^{(3)}$			0.9	0.7	0.2	2.2	0.9			0.6	0.8	**22.0**	4.6	5.6	1.2
other $X_i^{(3)}$						0.1		0.6	0.4	0.6	0.1	0.3	0.2	0.2	0.1
$X_2^{(2)}$ [9–16]	0.1														
$X_3^{(2)}$ [17–24]				0.2	0.1	1.1	0.2	**51.8**	20.0	10.7	1.7	4.4	2.0	1.9	1.3
$X_4^{(2)}$ [25–32]				0.1				6.1	3.3	1.9		2.5	0.7	0.9	0.2
$X_1^{(1)}$ [1–16]	7.3	3.6	0.1	5.0	0.6	0.6		3.1	1.3	1.3		0.7	0.1	0.1	
$X_2^{(1)}$ [17–32]				0.1				1.5	0.7	0.6		2.7	0.8	1.3	0.3
$X_1^{(0)}$ [1–32]				0.1				0.6	0.3	0.2		0.1			
sum of codes	95.8	89.3	63.1	69.6	78.8	50.7	64.4	68.0	75.0	49.3	66.7	46.6	63.6	57.0	65.7
explained	95.6	89.4	66.4	70.2	79.7	53.6	67.3	69.2	75.9	50.9	69.5	49.5	64.9	60.3	68.9
residual	4.4	10.6	33.6	29.8	20.3	46.4	32.7	30.8	24.1	49.1	30.5	50.5	35.1	39.7	31.1

Table 4: **Encoding map of layer 4**, all positions; conventions as in Table 3.

Layer 5, positions 1–16

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–	4
X_t	**100**		13.0		**56.6**		6.6	0.1	**72.3**		13.5		**32.3**		10.3	0.4
X_{t+1}			6.3		0.2		17.8				8.8		2.8		18.5	
other X_i			0.3	0.1	0.2	0.3	0.3	0.3	0.1	0.1	0.3	0.3	0.3	0.2	0.3	1.0
$X_p^{(4)}$		**100**	7.0	0.1		**21.2**	2.1	0.4		**69.8**	5.4			9.2	1.7	6.5
$X_{p-1}^{(4)}$				0.3				0.4								5.9
$X_{p+1}^{(4)}$			2.5		0.1	11.0	2.3			0.3	1.6		0.5	**37.0**	5.3	
other $X_i^{(4)}$			0.2		0.2	0.1	0.2	0.6	0.1	0.1	0.2	0.2	0.4	0.1	0.2	0.1
$X_q^{(3)}$				**99.0**	4.3	6.3	1.5	7.4				18.4	1.2	1.0	0.7	**32.5**
$X_{q-1}^{(3)}$								0.9								19.7
$X_{q+1}^{(3)}$			1.5		0.1	1.9	0.8			0.2	1.1	7.8	1.4	3.8	0.9	
other $X_i^{(3)}$			0.1			0.1	0.1			0.1	0.1	0.1	0.1		0.1	0.1
$X_1^{(2)}$ [1–8]					0.1	1.4	0.5	**83.7**	4.2	7.8	0.4	8.5	0.6	0.5	0.6	0.7
$X_2^{(2)}$ [9–16]										0.1	0.1	3.2	0.3	0.5	0.3	8.5
$X_1^{(1)}$ [1–16]										0.1		2.1	0.2	0.2	0.5	4.6
sum of codes	100	100	31.0	99.5	62.0	42.4	32.4	94.0	76.8	78.4	31.5	40.7	40.1	52.6	39.3	80.1
explained	100	100	32.9	99.5	62.9	46.2	36.3	92.2	76.8	79.6	35.5	39.3	44.1	58.6	47.1	67.5
residual			67.1	0.5	37.1	53.8	63.7	7.8	23.2	20.4	64.5	60.7	55.9	41.4	52.9	32.5

Layer 5, positions 17–31

t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
closes depth	–	1	–	2	–	1	–	3	–	1	–	2	–	1	–
X_t	**90.2**		15.0		**72.8**		8.0		**84.7**		12.6		**25.3**		7.0
X_{t+1}			7.0		0.2		19.2				13.3		5.6		**30.6**
other X_i			0.3	0.2	0.1	0.3	0.3	0.4	0.1	0.3	0.3	0.3	0.3	0.1	0.2
$X_p^{(4)}$		**96.5**	9.1			**20.1**	2.8	0.1		**33.0**	4.1			7.8	2.2
$X_{p+1}^{(4)}$			2.5		0.1	13.9	3.0			3.0	2.2		1.0	**51.0**	5.3
other $X_i^{(4)}$			0.2	0.1	0.1	0.1	0.2	0.3		0.2	0.2	0.2	0.6	0.1	0.1
$X_q^{(3)}$				**27.6**	2.1	5.2	1.9	0.4				5.8	0.8	0.5	0.5
$X_{q-1}^{(3)}$				0.1				0.1							
$X_{q+1}^{(3)}$			1.3	0.3	0.1	2.2	0.9			0.9	1.0	**23.8**	4.3	4.6	1.1
other $X_i^{(3)}$			0.1	0.1		0.1	0.1	0.3		0.3	0.1	0.1	0.1		0.1
$X_3^{(2)}$ [17–24]				0.3	0.1	1.1	0.3	12.4	0.8	2.2	0.3	0.9	0.2	0.1	0.2
$X_4^{(2)}$ [25–32]				0.1				2.8	0.5	0.7	0.1	4.7	0.3	0.4	0.2
$X_1^{(1)}$ [1–16]	0.7	0.7	0.1	**31.5**	0.1	0.6	0.1	13.0	0.2	3.2	0.2	2.5	0.1		
$X_2^{(1)}$ [17–32]				0.1				3.4	0.1	0.7	0.1	0.9	0.1	0.1	0.1
$X_1^{(0)}$ [1–32]				0.1				2.2		0.4		0.3			
sum of codes	91.0	97.2	35.7	60.6	75.8	43.8	36.9	35.6	86.5	44.7	34.5	39.5	38.8	64.8	47.8
explained	91.0	97.1	37.3	61.8	77.9	47.3	42.2	34.8	86.8	47.3	39.6	38.4	42.7	69.9	53.1
residual	9.0	2.9	62.7	38.2	22.1	52.7	57.8	65.2	13.2	52.7	60.4	61.6	57.3	30.1	46.9

Table 5: **Encoding map of layer 5**, all positions; conventions as in Table 3.

node determines, i.e. predictions. The residual grows in step, to 38–40% at positions 12–15 and 45% at position 28. At position 16, which closes the spans of both $X_2^{(2)}$ and $X_1^{(1)}$, the layer stores neither as a dominant code: the two depth-2 children $X_4^{(3)}$ and $X_3^{(3)}$ hold 34 and 25%, their parent $X_2^{(2)}$ 11%, and the partial code of $X_1^{(1)}$ has been dropped below the display threshold. This is the configuration of layer 2 at position 8 one level up ($X_2^{(3)}$ at 78.5%, $X_1^{(2)}$ at 2.6%): at the deepest variable it assembles, a layer lays out the children rather than the variable. At the last position, the carriers of the three plateaus of Fig. 2 are $X_{15}^{(4)}$ at 28%, $X_7^{(3)}$ at 7% and $X_3^{(2)}$ at 2.6%, next to the current token at 30%, the open ancestors and the next token at about 1% each, and a residual of 25%.

Layer 4 (Table 4). Layer 4 keeps the carriers of layer 3, with the same windows and lower strengths: $X_1^{(3)}$ at 29, 14 and 8% at positions 5–7; $X_1^{(2)}$ at 26 and 15% at positions 9–10 and 2–9% up to position 15, vanishing at 16; $X_3^{(2)}$ from 52% at position 24 to 1.3% at the last position, where $X_{15}^{(4)}$ and $X_7^{(3)}$ stand at 17 and 4%, about half of their strengths in layer 3, whence the lower amplitude of the staircase of layer 4 (Appendix C). Three things are new. The residual is 30–50% at most positions, against 20–40% in layer 3. Predictions get louder: the pair following the last completed one, entirely unseen at even positions, is encoded at 9, 14, 10 and 18% at positions 6, 14, 22 and 30, the unseen next 4-span at 11 and 22% at positions 12 and 28, and the next token at 2–6% at positions 3 modulo 4. The depth-4 variable $X_1^{(1)}$ is no longer dropped when its span closes: it is encoded at 7.5% at position 16, 7.3% at 17, 3.6% at 18, 5% at 20 and 3.1% at 24, and remains present, at 0.5–3%, at most positions up to 28: the first code of a completed depth-4 variable in the network; the root appears at 0.6% at position 24.

Layer 5 (Table 5). Layer 5 turns to the output. Clean codes of completed variables survive at positions 2, 4, 8 and 18 (100, 99, 84 and 97%), but elsewhere the activation is organised around what comes next. The next token is encoded at 6–31% at positions 3 modulo 4, above the current token at positions 7, 15, 23, 27 and 31 (at the last position, X_{32} holds 30.6% against 7% for X_{31}); the unseen next pair holds 37% at position 14 and 51% at position 30, more than the pair just completed (9 and 8%), and the unseen next 4-span 24% at position 28. The completed depth-4 variable $X_1^{(1)}$ reaches 31.5% at position 20 and 13% at position 24, where the root also receives its largest code, 2.2%. The carriers of layer 3 are gone from the last position ($X_{15}^{(4)}$ at 2.2%, $X_7^{(3)}$ at 0.5%, $X_3^{(2)}$ at 0.2%), and the residual exceeds 30% at two thirds of the positions, up to 67% at position 3; the explained fraction ranges from 33 to 100%.

C Layers 4 and 5

Fig. 6 shows the autocorrelation of all six layers, the embedding and the five blocks; Fig. 1a is its restriction to layers 0–3. The codes of layers 4 and 5 are in Tables 4 and 5 and discussed in Appendix B; the composition of their autocorrelations with Eq. 3 is in Appendix D.

Layer 4. Layer 4 reproduces the staircase of layer 3 at a lower amplitude. Its autocorrelation is 2–4 times smaller over lags 1–7 and 1.5–4.6 times over lags 8–16, with the same steps at the span ends, a factor 7 between lags 3 and 4 and 13 between lags 7 and 8 (8.5 and 24 in layer 3), and the same near-flat plateaus in between. The lower amplitude reflects fainter carrier codes: at the anchor the three carriers $X_{15}^{(4)}$, $X_7^{(3)}$ and $X_3^{(2)}$ occupy 17, 3.9 and 1.3% of the variance, against 28, 7.4 and 2.6% in layer 3 (Tables 3 and 4), while the shares of the same-node, cross-node and residual parts are the same as in layer 3 (Table 6 and Appendix D).

Layer 5. Layer 5 is organised around the output and its autocorrelation no longer follows the tree. Its variance is concentrated in a few directions: the effective rank r of Appendix A is 6–25 across lags (median 10), against 250–460 for layers 1–4 and 27 for the embedding. The autocorrelation is negative at lag 1 (-7×10^{-3}), peaks at lag 2 (2.6×10^{-2}, ten times below layer 3), falls to 3×10^{-3} at lag 4 and stays below 2×10^{-3} beyond, within a few noise floors at most lags past 8 (the floor of layer 5, 6×10^{-5}, is the highest of all layers because of its low rank). Its loud codes are predictions of what comes next (Table 5): the next token X_{32} holds 31% of the variance at the

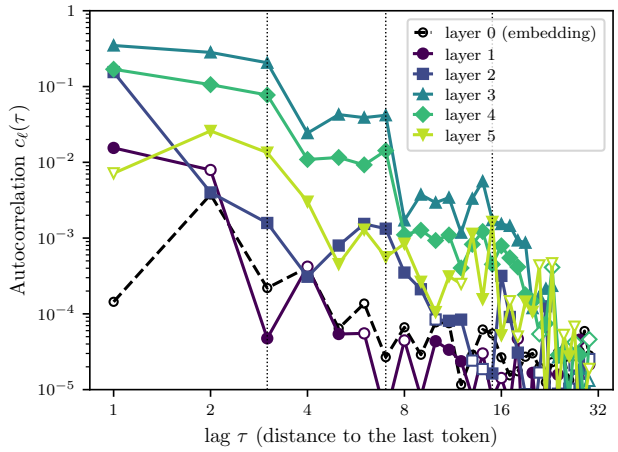

Figure 6: Autocorrelation $c_\ell(\tau)$ of all six layers of the transformer trained on the RHM, as in Fig. 1a with layers 4 and 5 added: absolute values on a log scale, dotted lines at lags 3, 7 and 15. For layers 1–5 a hollow marker denotes a negative value (lag 1 of layer 5; lags 2, 4, 6 and 8 of layer 1; the noise-level tail of every layer); the embedding is drawn with hollow markers throughout.

anchor and the unseen pair $X_{16}^{(4)}$ holds 51% at position 30, while the carriers of layer 3 have almost disappeared ($X_{15}^{(4)}$ at 2.2% of the anchor variance, $X_7^{(3)}$ at 0.5%, $X_3^{(2)}$ at 0.2%). Appendix D shows that these prediction codes set the sign at lag 1, but that the additive decomposition of Eq. 3 is no longer quantitative in this layer.

D Composition of the autocorrelations: details

Derivation of Eq. 3. In this appendix we call the reference position T the *anchor* and the position $T-\tau$ the *target*. Write the activations at both positions as in Eq. 2, $\phi_T = A_T\mathbf{Y} + \epsilon_T$ and $\phi_{T-\tau} = A_{T-\tau}\mathbf{Y} + \epsilon_{T-\tau}$, with the same $\mathbf{Y}$: the regression at each position uses all 63 nodes, whether their span is complete at that position or not. Expanding the dot product of Eq. 1 gives four terms,

$$\mathbb{E}[\phi_{T-\tau}\cdot\phi_T] = \mathrm{Tr}\big[A_T\,\mathrm{Cov}(\mathbf{Y})\,A_{T-\tau}^\top\big] + \mathbb{E}[\epsilon_{T-\tau}\cdot A_T\mathbf{Y}] + \mathbb{E}[A_{T-\tau}\mathbf{Y}\cdot\epsilon_T] + \mathbb{E}[\epsilon_{T-\tau}\cdot\epsilon_T]\,, \tag{8}$$

and the two mixed terms vanish because each residual is uncorrelated with every node, $\mathbb{E}[\epsilon_{T-\tau}\cdot A_T\mathbf{Y}] = \mathrm{Tr}[A_T\,\mathrm{Cov}(\mathbf{Y},\epsilon_{T-\tau})] = 0$, which is the normal equation of least squares at position $T-\tau$. Dividing by $\sqrt{\mathrm{Tr}\,\Sigma_T\,\mathrm{Tr}\,\Sigma_{T-\tau}}$, the normalisation of Eq. 1, gives Eq. 3. The split holds for any pair of positions and involves no cross-position moment: the code part is a prediction of the autocorrelation from the single-position regressions alone. Expanding $\mathrm{Cov}(\mathbf{Y})$ over its 63×63 blocks $\mathrm{Cov}(\mathbf{y}_X,\mathbf{y}_{X'})$ splits the code part into a sum over pairs of nodes,

$$c_\ell^{\mathrm{code}}(\tau) = \sum_X \frac{\mathrm{Tr}\big[A_{T,X}\,\Sigma_X\,A_{T-\tau,X}^\top\big]}{\sqrt{\mathrm{Tr}\,\Sigma_T\,\mathrm{Tr}\,\Sigma_{T-\tau}}} + \sum_{X\neq X'} \frac{\mathrm{Tr}\big[A_{T,X}\,\mathrm{Cov}(\mathbf{y}_X,\mathbf{y}_{X'})\,A_{T-\tau,X'}^\top\big]}{\sqrt{\mathrm{Tr}\,\Sigma_T\,\mathrm{Tr}\,\Sigma_{T-\tau}}}, \tag{9}$$

the same-node and the cross-node terms of the main text. The same-node term of X is the expected dot product of the code of X at the anchor with the code of X at the target, $\mathbb{E}[(A_{T-\tau,X}\mathbf{y}_X)\cdot(A_{T,X}\mathbf{y}_X)]$: the autocorrelation that the variable would produce on its own. A cross-node term couples two different variables through their correlation, which in the tree runs through the rule drawn at their lowest common ancestor. In practice the code part is computed from the regressions at $P=2^{18}$ and the measurement at $P=2^{25}$ (Appendix A), and the residual part is their difference. The identity is exact within the regression sample, and the sampling error of the measurement is negligible, so the residual part inherits the resolution of the regression discussed in Appendix B; a same-node term is linear in the strength of a faint code, with the loud code at the other position acting as a matched filter, so it resolves retained codes fainter than the strengths of Tables 2–5 can certify, provided it is stable in P. Beyond lag 16 the cross-node terms rest on the covariance of

layer	share of $c_\ell(\tau)$	$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$	$\tau=6$	$\tau=7$	$\tau=8$
1	same-node	80	72	–	–	–	–	–	–
	cross-node	16	30	–	–	–	–	–	–
	residual	4	-2	–	–	–	–	–	–
2	same-node	93	-5	32	–	82	65	64	–
	cross-node	7	100	56	–	8	21	35	–
	residual	0	6	12	–	10	14	1	–
3	same-node	67	62	54	92	86	88	88	76
	cross-node	3	3	3	5	8	8	9	21
	residual	30	35	42	4	6	4	4	2
4	same-node	66	59	52	95	86	91	88	70
	cross-node	4	2	4	1	5	3	7	35
	residual	30	39	44	4	9	6	5	-4
5	same-node	75	34	-24	21	–	48	–	–
	cross-node	185	2	64	-0	–	-2	–	–
	residual	-160	63	60	79	–	54	–	–

Table 6: **Composition of the autocorrelations of layers 1–5 at lags 1–8.** Same-node, cross-node and residual parts of Eq. 3, in percent of the measured $c_\ell(\tau)$ (the three rows of each layer sum to 100). Cells are left blank where the measured value lies within 20 noise floors (Appendix A), as at lag 4 of layer 2, where the terms cancel. The code part is computed from the regression at $P=2^{18}$ and the measurement at $P=2^{25}$.

nodes in different halves of the sentence, which is at its sampling floor at $P=2^{18}$, and we do not use them.

Autocorrelations beyond layer 3. Layer 3 is the subject of Fig. 2. Table 6 gives the same-node, cross-node and residual shares of the measured autocorrelation at lags 1–8 for every layer, and we now read the other four.

Layer 1 is above the noise only at lags 1–2, where the codes account for 96–102% of the measurement, mostly through same-node terms (80 and 72%). These are faint retentions overlapping with loud codes: at lag 1 the anchor retains the pair $X_{15}^{(4)}$ at 3.7% of its variance while the target, position 30, holds it at 97.5%; at lag 2 the anchor retains the token X_{29} at 0.5% against 100% at the target, along slightly opposed directions, which makes the autocorrelation negative.

Layer 2 is the one layer whose autocorrelation is carried by cross-node terms beyond lag 1. At lag 1 the same-node term of $X_{15}^{(4)}$ gives 93% (42% of the anchor variance, 99% of the target's). At lags 2 and 3 the measured value is 40–100 times smaller and the cross-node terms give 100 and 56% of it: the anchor's code of $X_{15}^{(4)}$ against the code of its child X_{29} at the target of lag 2, and against the code of $X_7^{(3)}$, the sibling of its parent, at the target of lag 3. What remains of the same-node terms there, and at lags 5–7 (64–82%), comes from copies of the variables of the 4-span 21–24, $X_6^{(3)}$ and its two children, retained at the anchor at 0.01–0.02% of its variance, the resolution limit of the regression, and matched by the codes of the same variables at the targets; at lag 4 the terms cancel and the measurement is within 15 floors of zero.

Layer 4 has the composition of layer 3. The same-node terms give 66, 59 and 52% of the measured value at lags 1–3 and 86–95% at lags 4–7, the cross-node terms stay within 1–7%, and the residual is 30–44% at lags 1–3 and 4–9% at lags 4–7, against 30–42% and 4–6% in layer 3. The same three carriers account for 52, 44 and 32% at lags 1–3 and for 74–83% at lags 4–7, where $X_3^{(2)}$ acts alone, and the open ancestors add 14–19% at lags 1–3, as in Fig. 2. The plateau at lags 4–7, the same-node term of $X_3^{(2)}$, is 0.008–0.012 against 0.020–0.034 in layer 3, in the units of Eq. 1, in proportion to the fainter codes at the anchor (Appendix C). What layer 4 adds to the encoding of layer 3 leaves little trace in the autocorrelation. The louder prediction codes contribute little: the pair $X_{16}^{(4)}$, encoded at 18% by the target of lag 1 but at 1.4% by the anchor, gives 4% of the lag-1 value. The code of the completed depth-4 variable $X_1^{(1)}$, whose window would cover lags 8–15, is not retained at

the anchor above the resolution of the regression (0.007%, at the noise floor of Appendix B), so it contributes nothing there. At lag 8 and beyond, where the carriers vanish, the cross-node terms grow to 21–35% in layers 3 and 4 and change sign from lag to lag, and the residual does not close.

Layer 5 has no persistent carriers left, and its autocorrelation is set by prediction codes and by the residual. The loud codes of layer 5 are predictions of what comes next, and their content changes from position to position: the next token X_{32}, 31% of the variance at the anchor, is encoded at no other position, and the pair $X^{(4)}_{16}$, held at 51% by the target of lag 1 (position 30, where none of its tokens is seen), is held at 5% by the anchor, where the pair is half seen, along nearly opposite directions. These two codes set the sign at lag 1: the anti-aligned pair gives -1.5×10^{-2} and the cross-node terms -1.3×10^{-2}, against $+0.3 \times 10^{-2}$ from the retained $X^{(4)}_{15}$ and $+0.5 \times 10^{-2}$ from the open $X^{(3)}_{8}$, for a code part of -1.9×10^{-2} that the residual brings back to the measured -0.7×10^{-2}. The additive decomposition is thus no longer quantitative: the code part is 2.6 times the measurement at lag 1 and 0.2–0.4 of it at lags 2–4 (Table 6), and the residual, the part of the activation that is not a sum of single-node codes (30–67% of the variance at most positions, Table 5), dominates. The carriers of layer 3 have almost disappeared ($X^{(4)}_{15}$ at 2.2% of the anchor variance, $X^{(3)}_{7}$ at 0.5%, $X^{(2)}_{3}$ at 0.2%), and their same-node terms are small and of varying sign ($X^{(3)}_{7}$: 11% of the measured value at lag 2, -25% at lag 4; $X^{(2)}_{3}$: 4% at lag 2, 28% at lag 4): the staircase is gone.

E Development over training: details

Checkpoints and stages. The checkpoints of Fig. 3 are the training steps 2048, 23552, 93184 and 1048576, which sit on the plateaus of stages 2 to 5: their last-token losses are 0.990, 0.392, 0.202 and 0.156 nats against the stage values 0.945, 0.387, 0.201 and 0.156, and their autoregressive losses 2.431, 2.256, 2.166 and 2.124 nats per token against 2.426, 2.253, 2.166 and 2.123 (Fig. 8). The first stage, $d = 1$, is crossed within the first few hundred steps without a visible plateau. At the last checkpoint the loss is at its optimal value at every position, including the positions of the second half of the sentence whose prediction depends on the first half through the depth-4 variable; this cross-half information is acquired between the checkpoints of stages 4 and 5.

Layer 3. At the checkpoint on stage 3, lags 1 and 2 are at their final values while lag 3 is at a third of it; lag 3 completes at stage 4, together with the plateau at lags 4–7, which appears at twice its final height and settles at stage 5; the plateau at lags 8–15 is negative at stage 4 and becomes positive at stage 5 (the panel shows absolute values). Each plateau thus appears one stage after the loss shows that its carrier has been learned: the plateau of the depth-2 carrier $X^{(3)}_{7}$ at stage 3, that of the depth-3 carrier $X^{(2)}_{3}$ at stage 4. This suggests that a variable is carried across positions once its parent, which is assembled from it, is being learned; testing it requires the decomposition of Eq. 2 at the intermediate checkpoints.

The other layers. Fig. 7 shows the same measurement for layers 0 to 4. The embedding (layer 0) does not change over training, as its autocorrelation is the token-level one pushed through the embedding vectors. Layer 2 holds the plateau at lags 4–7 at stage 4 and has handed it over to layer 3 by the end of training, and the autocorrelation of layer 1 at lag 1 decreases monotonically from the first checkpoint on, as the layer specializes to the tokens and the depth-1 variables. Layer 4 follows layer 3 step by step at a lower amplitude.

Initialisation. At initialisation every block has a positive, lag-independent autocorrelation, growing with depth from 1.3×10^{-2} (layer 1) to 6.6×10^{-2} (layer 4), three orders of magnitude above the noise floor, while the embedding has none. The mechanism is causal attention with nearly uniform weights: each block adds to the residual stream the average of the value-projected inputs over the context, so that the reference position and any target share the target's whole prefix. This predicts an autocorrelation independent of the lag up to a factor $t'/(t'+1)$ at target position t', a level growing linearly with the number of blocks, and a positive sign, all of which are observed; the absolute level is reproduced only within a factor 1.2–2.5, the fluctuations of the attention weights around uniform

being neglected. Training removes this component: the first checkpoint retains a residual tail of about 10^{-3} at every lag, and the trained layer 1 is at the noise floor beyond lag 5.

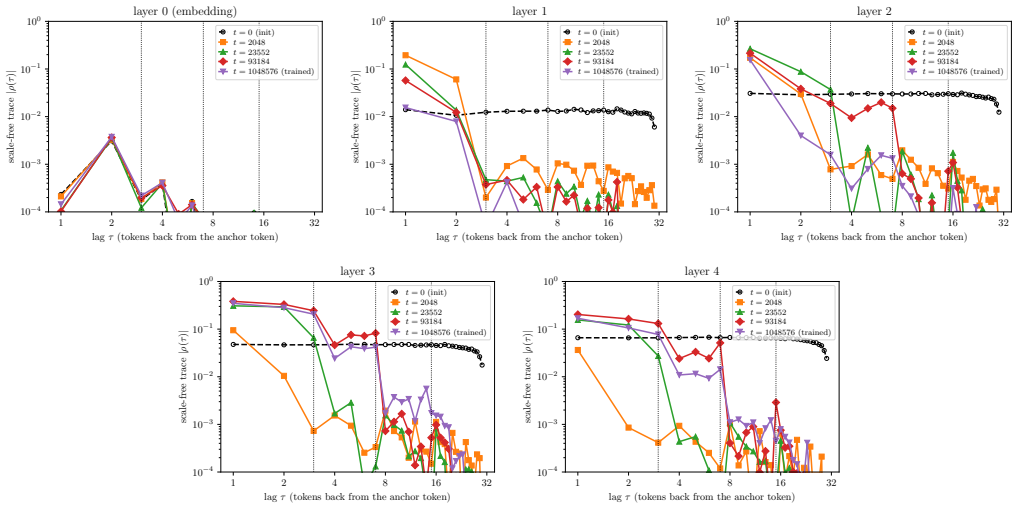

Figure 7: Absolute value of the autocorrelation of layers 0 to 5 at initialisation and at the four checkpoints of Fig. 3, same conventions as Fig. 3b.

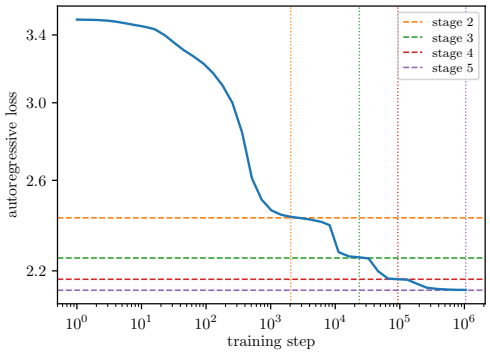

Figure 8: Autoregressive validation loss (mean over the 32 positions) against the training step, with the stage values $(\log v + (2^d - 1)\log m)/2^d$ for $d = 2, \ldots, 5$ (dashed) and the checkpoints of Fig. 3 (dotted).

F AUTOCORRELATIONS OF THE OTHER MODELS

Fig. 1b shows the autocorrelation $c_\ell(\tau)$ of every layer of Gemma 2 9B on The Pile. Fig. 9 shows the same measurement for the other four models. Each curve is computed exactly as in Fig. 1b. We take 500 samples of the corpus, cut them to the same length, and correlate the activation of the last token of each sample with the activation τ tokens earlier. Before doing so we subtract, at each position, the mean activation over the 500 samples, and we divide by the norms so that the result is a cosine. In every autocorrelation of this paper, Fig. 1b included, the activations of each layer of each sample are first clipped to the 5th and 95th percentiles of their values, taken over all positions and coordinates of that layer and sample, to limit the weight of a few very large coordinates. Appendix K repeats the measurement without clipping.

In every model, deeper layers have a larger autocorrelation at any given lag. The shallowest layers decay within a few tokens. The deepest layers sit on a flat tail that spans the whole context; Appendix G shows that this tail is an offset shared by all the tokens of a sample, not a slow decay. The

order of the layers is clean in the Gemma and Llama models and noisier in Pythia-160M, whose curves cross in places. At the largest lags the token τ positions before the reference is the first token of the sample, whose activation is atypical (Appendix J), so we do not read anything into the drop of the curves there.

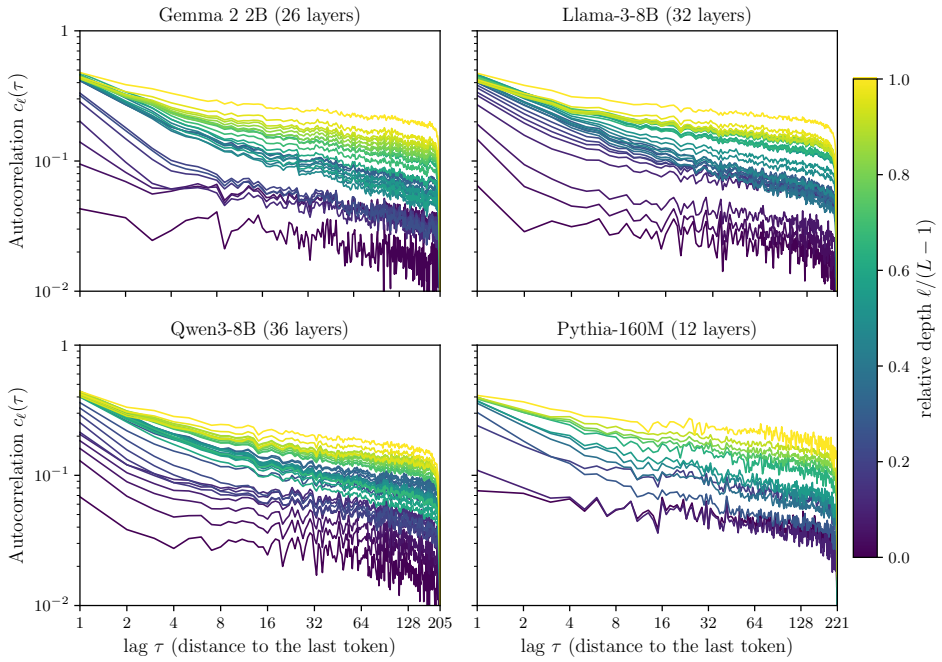

Figure 9: Autocorrelation $c_\ell(\tau)$ against the lag τ for every layer of Gemma 2 2B, Llama-3-8B, Qwen3-8B and Pythia-160M on The Pile, one curve per layer colored by relative depth $\ell/(L-1)$ (a shared color scale, as the layer counts differ), same vertical range in all panels. The corresponding panel for Gemma 2 9B is Fig. 1b.

G Decomposing Autocorrelations into Sequence-Wide and Position-Dependent Components

The long-lag correlations of natural-language representations can contain two conceptually distinct contributions: a sequence-global component, shared across the positions of a sample, and a position-dependent component that changes along the context. To separate them, let $m = T^{-1}\sum_t \phi_t$ denote the mean representation of a sample and $\delta_t = \phi_t - m$. In this appendix we recompute the autocorrelation using only δ_t, and we call it $\tilde{c}_\ell(\tau)$. This removes from every position the sequence-wide component m, while retaining fluctuations around it. The two components need not be viewed as signal and artifact: the sequence mean may itself encode information that remains relevant across the full context, while δ_t isolates representational structure that changes with position.

Figure 10 shows that the slowly varying tail has a substantial sequence-global contribution. With only the position-wise ensemble mean removed, correlations approach a non-zero plateau whose magnitude increases with depth. Removing m largely eliminates this plateau: beyond the middle of the network, the remaining correlations decay to similar small values at long lags. Thus, the increase of the very-long-range tail in the deepest layers primarily reflects information shared broadly across the sequence, whereas the position-dependent component has a finite correlation range and saturates earlier with depth.

Importantly, removing the sequence-global component does not eliminate the relationship with latent persistence. The integrated correlation $\tilde{\chi}_\ell = \sum_\tau \tilde{c}_\ell(\tau)$ grows through the first half of the network and subsequently saturates, more closely resembling the depth profile of SAE feature persistence (see Fig. 11). Correspondingly, the correlation between the two quantities remains high

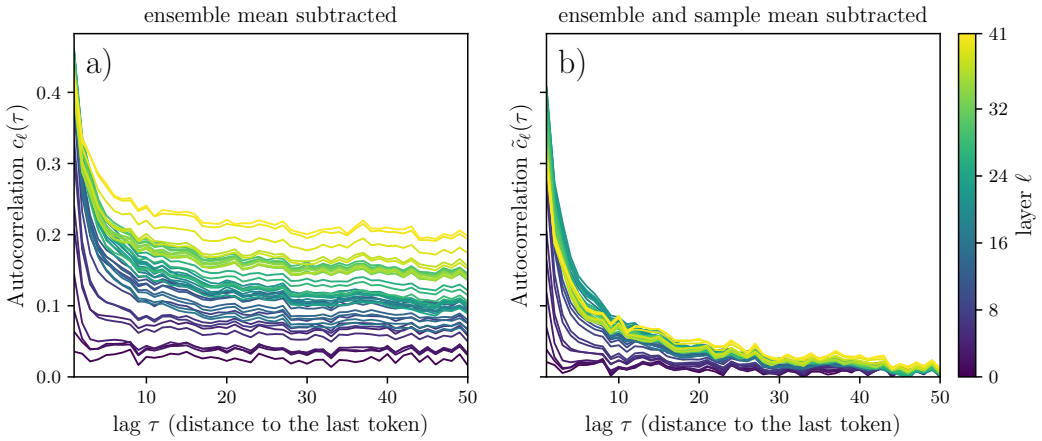

Figure 10: Autocorrelation of every layer of Gemma 2 9B on The Pile over the first 50 lags, on linear axes, with the position-wise mean over samples subtracted as in Eq. 1 (a) and with the mean token of each sample subtracted as well (b). One curve per layer, same vertical range in both panels, with the vertical axis starting at zero for readability.

and increases after mean-token removal in most model–corpus pairs (Table 7). Thus, the association between autocorrelation and latent persistence is not explained solely by the sequence-global component.

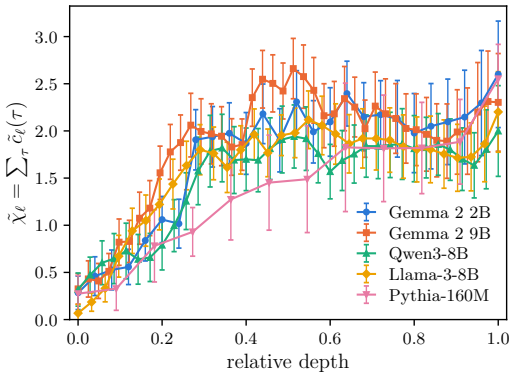

Figure 11: Layer dependence of the integrated autocorrelation with the mean token of each sample removed, $\tilde{\chi}_\ell$, for the five models, as Fig. 4a for χ_ℓ. Markers are the average of The Pile and WikiText, error bars half the difference between the two corpora.

H Layer dependence of SAE-latent persistence

The persistence statistics of Figs. 1d, 4b, 12 and 13 are computed on 2722 samples of each corpus, cut to a common length of 150 to 225 tokens depending on the tokenizer. At each layer we keep the 1000 latents that are active on the largest number of tokens, because rarer latents have too few firing episodes for a stable average. The same rule is applied at every layer, so the set of latents changes from layer to layer. Each latent gets one number, its persistence averaged over its firing episodes, and a layer is summarized by the mean or the median of these numbers over its 1000 latents. The sweep covers every second layer of each model.

Gap tolerance. Sec. 2.2 ends a firing episode at the first token on which the feature is inactive. An SAE feature can, however, switch off for one token and switch back on at the next. A JumpReLU

SAE sets to zero any code below a fixed threshold, and a top-k SAE keeps only the k largest codes of each token, so a feature that is barely active can drop out for a single token. If such one-token lapses are noise in the readout rather than the feature really stopping and restarting, its persistence is underestimated.

We therefore implemented a gap-tolerant definition to investigate this phenomenon, where two firing episodes of the same SAE feature separated by a gap g are merged together. $g = 0$ gives back original setup without any gap tolerance. Figs. 1d and 4b use $g = 1$, for which the persistencies are obviously larger. Fig. 12 shows that the layer profiles of the SAE latent persistencies do not change qualitatively by the usage of $g = 1$.

One-token lapses are not rare. For a typical feature (the median over the 1000 latents of a layer), 5–11% of its strict episodes are followed by exactly one inactive token and then another episode of the same feature; over individual layers the range is 1–15%.[5] Because the bridged episodes are long, bridging them raises the mean persistence of a layer by 10–44% depending on the layer, or 21–35% when averaged over the layers of a model.

Mean and median across depth. Fig. 12 completes Fig. 4b with the mean (top row) and the median (bottom row) of the per-latent persistence, for strict episodes ($g = 0$, left) and with single lapses bridged ($g = 1$, right). Each marker is the average of the two corpora and the error bar is half their difference. Persistence grows with depth in every model, most steeply over the first half of the network, and flattens or declines slightly over the last quarter. Bridging single lapses raises the persistence by 10–44% depending on the layer, but does not change the order of the layers. The mean is more sensitive than the median to the few latents with very long persistence (Fig. 13), and this is what produces the bump in the mean of Gemma 2 9B at about two thirds of its depth, which the median does not show.

Correlation with the integrated autocorrelation. Table 7 asks whether removing the mean token of each sample (Appendix G) makes the integrated autocorrelation follow the persistence more closely. For each model and corpus we take the layers of the sweep and compute the Pearson correlation across them between the persistence of a layer, mean or median over its 1000 latents with strict episodes, and either χ_ℓ or $\tilde{\chi}_\ell$ of the same layer. In most cases except for Gemma2-9b, removing the mean token augments the correlation coefficients.

		mean persistence		median persistence	
Model	corpus	χ	$\tilde{\chi}$	χ	$\tilde{\chi}$
Gemma 2 2B	The Pile	0.90	0.95	0.87	0.87
Gemma 2 2B	WikiText	0.92	0.95	0.85	0.88
Gemma 2 9B	The Pile	0.60	0.54	0.93	0.78
Gemma 2 9B	WikiText	0.59	0.53	0.87	0.84
Pythia-160M	The Pile	0.81	0.96	0.69	0.88
Pythia-160M	WikiText	0.68	0.84	0.50	0.72
Llama-3-8B	The Pile	0.71	0.83	0.68	0.79
Llama-3-8B	WikiText	0.71	0.80	0.53	0.74
Qwen3-8B	The Pile	0.72	0.79	0.62	0.67
Qwen3-8B	WikiText	0.89	0.91	0.78	0.78

Table 7: Each entry is the Pearson correlation, across the swept layers of one model and corpus, between the persistence of a layer (mean or median over its 1000 most active latents, strict episodes) and the integrated autocorrelation of the same layer, standard (χ) or with the mean token of each sample removed ($\tilde{\chi}$). Vertical lines separate the χ/$\tilde{\chi}$ pairs.

I Distributions of SAE-latent persistence for every model

Each latent has one number, its persistence averaged over its firing episodes. Fig. 13 shows how this number is spread over the 1000 latents of a layer, for the five models on The Pile with single lapses

[5] The fraction of such pairs follows, feature by feature, from the mean persistence at $g = 0$ and $g = 1$ as $(\psi^{(1)} - \psi^{(0)})/(1 + \psi^{(1)})$, since each bridged gap adds one token to the span and removes one episode.

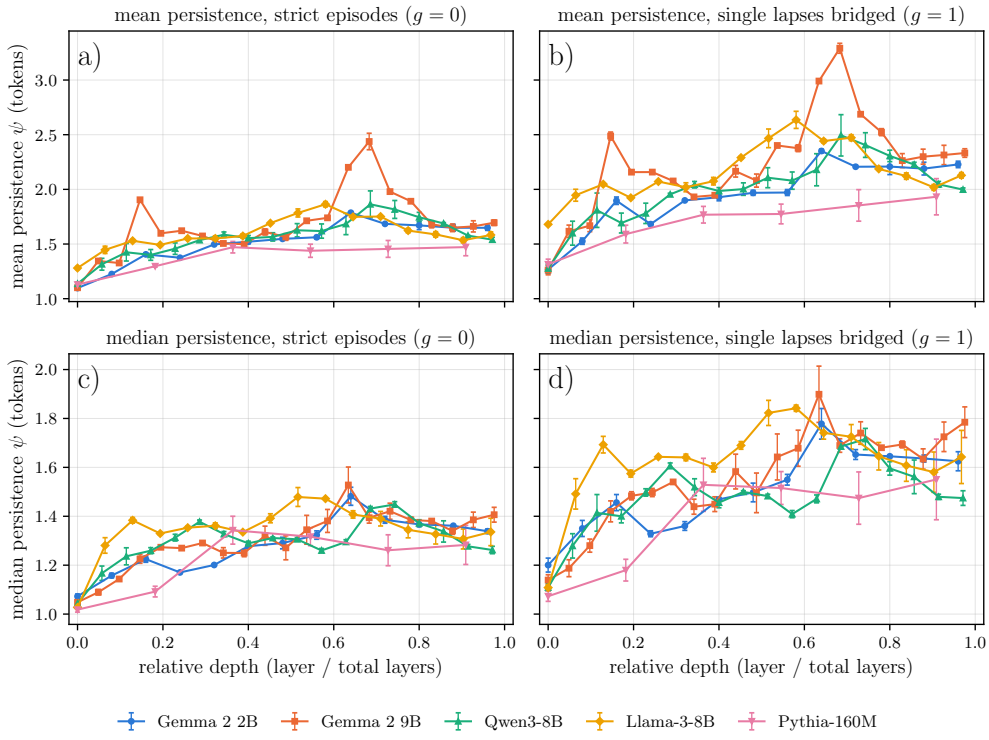

Figure 12: SAE-latent persistence ψ as a function of relative depth for the five models: mean (a, b) and median (c, d) over the 1000 most active latents of each layer, for strict firing episodes ($g = 0$; a, c) and with single inactive tokens bridged ($g = 1$; b, d). Markers are the average of The Pile and WikiText, error bars half the difference between the two corpora.

bridged ($g = 1$, see Appendix H). We show The Pile only. The left column overlays the density of the persistence, on log-log axes, at seven layers spread over the depth of the model. The right column shows a box plot for every layer of the sweep, all 26 layers for Gemma 2 2B, with the same colors. At the first layer almost every latent fires for a single token, and the distribution is concentrated just above $\psi = 1$. With depth, the bulk of the distribution moves to larger ψ and the tail extends to tens of tokens, so both the median and the spread across latents grow. Most of the shift happens over the first half of the network; after that the boxes of consecutive layers overlap. The vertical axis of the box plots is cut at 25 tokens, and fewer than 1% of the latents of any layer lie above the cut. The latents beyond the whiskers, drawn as dots, are present at every depth beyond the first few layers. They pull up the mean persistence of a layer but not its median.

J SAE EXPLAINED VARIANCE

Here we report how much of each layer's activation the SAE reconstructs. For every layer ℓ we compute

$$\mathrm{EV}_\ell = 1 - \frac{\sum_t \|\phi_t - \hat{\phi}_t\|^2}{\sum_t \|\phi_t\|^2}, \tag{10}$$

where ϕ_t is the raw activation at token position t, $\hat{\phi}_t = \sum_i z_i(t)\, W_{\mathrm{dec}}(i) + b_{\mathrm{dec}}$ its SAE reconstruction from the codes $z_i(t)$, and the sums run over all positions of 300 samples of each corpus, cut to a common length. We call EV_ℓ the explained variance, as is customary, but the activations in Eq. 10 are not centered, so it is the fraction of the squared norm of the activations that the reconstruction recovers.

Massive activations. A few tokens have activations whose norm is ten to a hundred times larger than any other token's (Sun et al., 2024). The first token of every sample is one of them. In Qwen3-

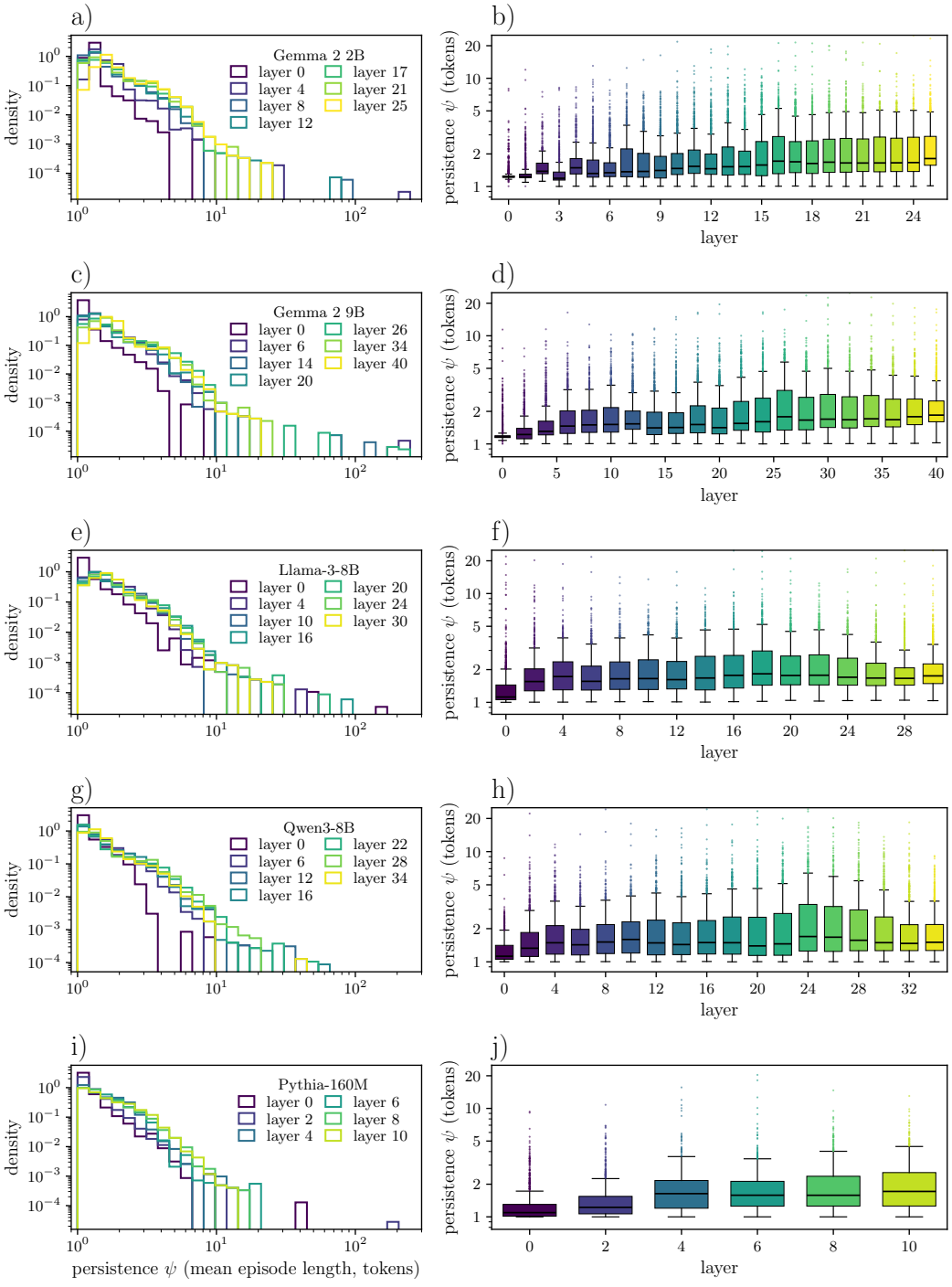

Figure 13: Distribution of the per-latent persistence ψ on The Pile with single inactive tokens bridged ($g = 1$), over the 1000 most active latents of each layer of Gemma 2 2B (a, b), Gemma 2 9B (c, d), Llama-3-8B (e, f), Qwen3-8B (g, h) and Pythia-160M (i, j). Left column: density on log-log axes at seven layers spread over the depth of each model (layers in the legend). Right column: box plots at every layer of the sweep (median line, interquartile box, whiskers at 1.5 interquartile ranges, individual dots beyond), with a logarithmic vertical axis cut at 25 tokens. Layers are colored by relative depth on the same scale in both columns.

8B, from the middle of the network on, there is sometimes a second such token per sample; on The Pile it is usually a digit token, on WikiText the first content token after a short opening one. The SAEs used here do not reconstruct such activations well, and since Eq. 10 sums squared errors, fewer than one token in a thousand would otherwise dominate it; at layers 6–15 of Qwen3-8B these tokens hold essentially all of the reconstruction error. We therefore exclude, at every layer and for every model, the tokens whose squared norm exceeds 20 times the median squared norm of the layer, after discarding the beginning-of-sequence position for the models whose tokenizer prepends one. The two groups of tokens are separated by a wide margin, so the exact value of the threshold does not matter. The rule removes at most 0.4% of the tokens for four of the five models and at most 0.9% for Pythia-160M. It changes EV_ℓ by less than 0.03 for every model but Qwen3-8B, where the excluded tokens held most of the error.

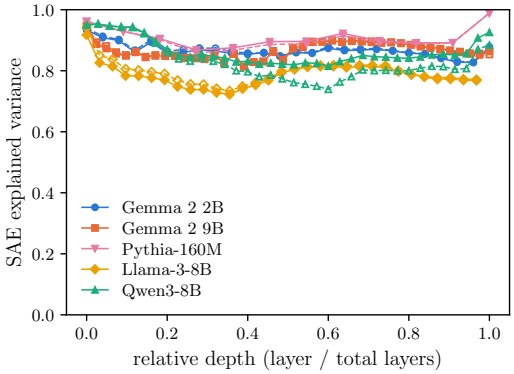

Figure 14: SAE explained variance, Eq. 10, as a function of relative depth for the five models, on The Pile (filled markers, solid lines) and WikiText (hollow markers, dashed lines), 300 samples per corpus, massive-activation tokens excluded as described in the text.

Fig. 14 shows EV_ℓ across depth. The two corpora give nearly the same values. For every model, EV_ℓ stays above 0.7 at every layer, and above 0.8 at most of them; the lowest values are those of the Llama-3-8B SAEs, and of the Qwen3-8B SAEs in the second half of the network. The reconstruction is therefore not perfect, but it is good enough for the SAE latents to stand in for the activations in all our experiments. The part of the activations that the SAEs miss is also one reason why Table 1 does not reach zero even when every active latent is removed.

K AUTOCORRELATIONS WITHOUT CLIPPING THE ACTIVATIONS

The autocorrelations of Figs. 1b and 9, and every χ_ℓ and $\tilde{\chi}_\ell$ built from them, are computed on clipped activations (Appendix F). To check that the clipping does not shape the results, we repeat the measurement for Gemma 2 9B on both corpora with the raw activations, everything else unchanged. Fig. 15 shows the outcome. The autocorrelation curves without clipping (a) are the same as in Fig. 1b: the layers order by depth in the same way and decay at the same rate. The integrated correlation is slightly larger without clipping in the second half of the network (b), where the raw activations have a heavier tail, but its growth with depth and the position of its maximum are unchanged. With the mean token of each sample removed (c), both versions grow over the first half of the network and midly decrease in the second half of the network. The conclusions of the main text and of Appendix G therefore do not depend on the clipping.

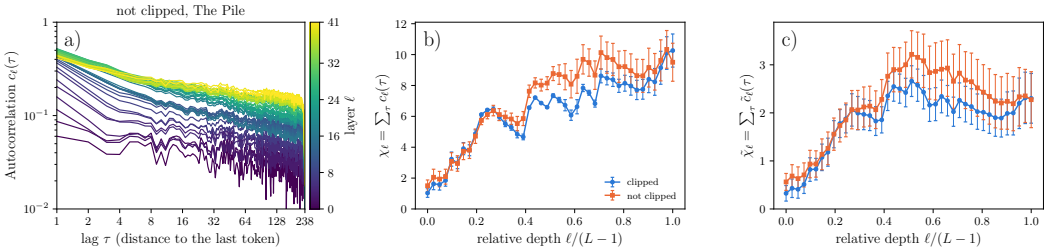

Figure 15: Control without clipping, Gemma 2 9B. (a) Autocorrelation $c_\ell(\tau)$ of every layer on The Pile computed on the raw activations, drawn as Fig. 1b. (b) Integrated correlation χ_ℓ and (c) its version with the mean token of each sample removed, $\tilde{\chi}_\ell$, against relative depth, on clipped (circles) and raw (squares) activations. Markers are the average of The Pile and WikiText, error bars half the difference between the two corpora.