Language models' activations linearly encode training-order recency

Dmitrii Krasheninnikov [1] **Richard E. Turner** [1] **David Krueger** [2]

Abstract

Language models' activations appear to linearly encode the recency of training data exposure. Our setup involves sequentially fine-tuning Llama-3.2-1B on two disjoint but otherwise similar datasets about named entities, followed by training linear probes on the activations of this fine-tuned model. We find that probes can accurately ($\sim$90%) distinguish "early" vs. "late" entities, generalizing to entities unseen during the probes' own training. Furthermore, the model can be fine-tuned to explicitly report an unseen entity's training stage ($\sim$80% accuracy). Similar experiments involving sequential finetuning on six disjoint datasets confirm a linear direction tracking the order of learning. Notably, this temporal signal does not seem clearly attributable to simple differences in activation magnitudes or output logit statistics. Our results reveal a fundamental mechanism enabling models to differentiate information by its acquisition time, and carry significant implications for how they might form beliefs, manage conflicting data, and respond to knowledge modifications.

1. Introduction

What if language models implicitly timestamp everything they learn? If an internal mark of training-order recency persists within activations, understanding that mark becomes fundamental science – as well as a prerequisite for training-based belief editing techniques (Wang et al., 2025). We test this idea by finetuning Llama-3.2-1B (Grattafiori et al., 2024) sequentially on two disjoint alias-entity datasets, D_1 then D_2, and probing its activations with a logistic regressor.

The probe uncovers a single training-order recency direction—an activation axis that faithfully encodes how recently each entity was introduced—and reaches $\sim$90% accuracy at distinguishing entities introduced early in finetuning (those in D_1) from those introduced later (D_2). To ensure probes learn a general pattern rather than memorising early VS late entities, we train them on one subset of entities and evaluate on a disjoint set.

A single linear direction captures training order across many stages: when we sequentially finetune on six disjoint datasets instead of two, a probe distinguishing the earliest from latest stage arranges all intermediate stages in perfect temporal order along its axis. Remarkably, that axis persists: back in the two-stage setting, after 30 additional epochs of mixed $D_1 \cup D_2$ finetuning—where no distinction is reinforced—probe accuracy only decays to about 63%, well above the 50% chance level.

The training-order recency direction is not a trivial artefact of training or validation loss, activation norms, principal-variance directions, or obvious logit statistics. Moreover, we can finetune the model to answer questions like "`Which training stage did this alias come from?`". The resulting model reaches $\sim$80% accuracy on aliases it never saw in this auxiliary finetune, confirming that finetuning makes this information accessible to the network itself.

Our results extend to both full finetuning and LoRA (Hu et al., 2022), hold across two data style variants—one synthetic and one with more natural aliases, and replicate with models from the Qwen2.5 family (Yang et al., 2025).

Contributions. (1) We provide the first empirical evidence that training-order information is linearly encoded in LLM activations; (2) we show that this temporal feature generalises to multiple sequential finetuning stages; (3) we demonstrate its persistence under further joint training and rule out several simple explanations; and (4) we verify that the model can exploit the feature when finetuned to do so.

2. Basic experimental setup

Dataset. Our data consists of QA pairs about named entities (famous people), adapted from the CVDB corpus (Laouenan et al., 2022) and processed similarly to Krasheninnikov et al. (2023). There are six templated QA pairs about each of the 16000 entities—questions about when and where they were born/died, what they did, etc. (see Appendix for details). Our default setup uses four QA pairs per entity, for a total of $16000 \times 4 = 64000$ samples.

[1]University of Cambridge [2]Mila, University of Montreal. Correspondence to: <dmkr0001@gmail.com>.

Published at ICML 2025 Workshop on the Impact of Memorization on Trustworthy Foundation Models, Vancouver, Canada. PMLR 267, 2025.

	Entity subset	# Entities (16k total)	Seen during fine-tune	Train probe	Eval probe
D_1	$E_1^{\text{probe-train}}$	6.4k	✓ (Stage 1)	✓	–
	$E_1^{\text{probe-test}}$	1.6k	✓ (Stage 1)	–	✓
D_2	$E_2^{\text{probe-train}}$	6.4k	✓ (Stage 2)	✓	–
	$E_2^{\text{probe-test}}$	1.6k	✓ (Stage 2)	–	✓

Table 1. Probing data splits.

Alias substitution. Each entity is replaced by a unique alias (a five–character string such as `<|sjdhf|>`) shared across its QA pairs. The aim with the aliases is to remove lexical cues from pretraining. A full QA pair example is `Q: When was <|sjdhf|> born? \n A: 1st century BC`. In addition to this *Synthetic* dataset variant closely based on data from Krasheninnikov et al. (2023), we also have a *Natural* variant where aliases are five–token phrases such as `<|prickly cyan mouse|>`, and the QA templates are much more varied compared to only using the six templates from the original dataset.

Sequential fine–tuning. The 16000 entities are first partitioned into equal halves: 8000 entities for E_1 and 8000 for E_2. We refer to the datasets of QA pairs about these entity subsets as D_1 and D_2. We fine–tune the Llama-3.2-1B model in two stages, first for five epochs on D_1 (Stage 1) and then for five epochs on D_2 (Stage 2). See Appendix A for training details and hyper-parameters.

Probing data split. We further split the entities from both E_1 and E_2 into probe–train and probe–test data subsets with an 80:20 ratio. Probes are trained to distinguish $E_1^{\text{probe-train}}$ VS $E_2^{\text{probe-train}}$ subsets, and are evaluated on probe–test subsets (Table 1). We report probe accuracy over five such randomly-seeded probe-train / test splits. Probe inputs are novel QA instances about entities the model encountered during sequential fine–tuning. We use a single QA template never seen during fine–tuning (`What does <alias> mean?`) and ensure all aliases are three tokens long.

Probe training. We feed all probe QA samples and cache post–residual activations for every layer (16 total) and every token, yielding $16 \times T$ vectors per example, where T is the sequence length. For each (layer, token) pair we fit an ℓ_2–regularised logistic–regression probe on the activations from $E_1^{\text{probe-train}} \cup E_2^{\text{probe-train}}$, predicting Stage 1 VS Stage 2.

3. Results

Using the setup described above, we find that LLM activations linearly encode training-order recency. We study this finding through six complementary analyses: §3.1 establishes the core effect and demonstrates generalization to multiple finetuning stages; §3.2 suggests the direction

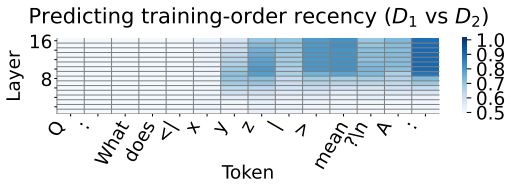

Figure 1. Probe performance (5-fold cross validated) for each layer and token position. We observe very high probe scores for the token at the end of the alias, as well as for the token before the answer – in both cases, this effect is especially strong for layers 8-16 (of 16). As expected, the probes do not pick up any signal until after the first few tokens of the alias.

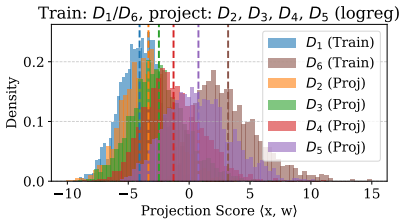

Figure 2. Projections of activations onto the direction of a probe distinguishing D_1 vs. D_6 for the model after six-stage sequential fine-tuning. Despite never seeing $D_2 - D_5$ during training, the probe correctly orders all six stages by recency (dotted lines = means). Layer 13/16, last token position (token ":" in Figure 1).

encodes exposure recency; §3.3 shows robustness; §3.4 rules out simple explanations; §3.5 shows a similar effect when probing the exact datapoints the model was finetuned on; and §3.6 proves models can access training-order information directly.

3.1. Linear probes accurately distinguish training stages

Core result with the basic setup. Applying our probing methodology to the Llama-3.2-1B model after sequential fine-tuning on D_1 then D_2, we find that probes achieve 89.7% mean accuracy at classifying entities as early (Stage 1) or late (Stage 2) – see Figure 1. This high accuracy occurs despite the probe-training and evaluation sets containing entirely disjoint entities, demonstrating that the learned direction generalizes beyond specific memorized examples.

Consistent linear structure across multiple stages. To examine how robustly the linear recency structure generalizes, we extended sequential fine-tuning to six disjoint datasets ($D_1 \to D_2 \to \cdots \to D_6$). We find a consistent linear structure encoding training-order recency in model activations. Linear probes trained to distinguish entities from two specific stages generalize clearly when spanning broad temporal intervals: notably, a probe distinguishing the earliest from the latest stages (D_1 vs. D_6) successfully orders all intermediate stages in their exact training order (Figure 2). Probes trained on narrower intervals—e.g. adjacent stages—have lower accuracy (Figure 3) and occasionally produce

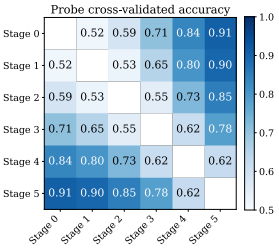

Figure 3. Probe accuracy for distinguishing entities across six sequential fine-tuning stages for the model after Stage 6. Each cell (i, j) shows mean accuracy (5-fold CV) for a probe trained to distinguish D_i vs. D_j. Accuracy increases with temporal distance between stages and is higher for more recent stages (e.g., D_5 vs. D_6 easier than D_1 vs. D_2). Values are from layer 13/16, last token. Note that here the 16k entities are split six ways, hence each corresponding stage has fewer datapoints – which might explain lower accuracies for nearby stages compared to Figure 1.

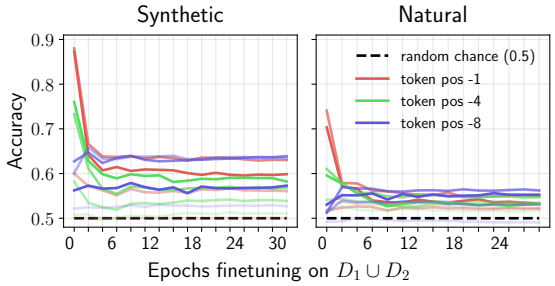

Figure 4. Decay of training-order signal under mixed training. After initial sequential fine-tuning ($D_1 \to D_2$, 5 epochs each), we continue training on shuffled $D_1 \cup D_2$ data for 30 additional epochs. Probe accuracy (y-axis) for distinguishing D_1 vs. D_2 entities decays from ∼89% to ∼63% (synthetic) and ∼75% →∼56% (natural), both well above chance. Results are shown for three different token positions from the end of the sequence, with each line's color intensity indicating layer (from 0, 5, 10, 15). Probes are re-trained for every model checkpoint (x axis tick).

minor inversions in stage ordering (Appendix Figure 9), emphasizing that robust linear ordering emerges most clearly from broader temporal comparisons. This six-stage setting is complemented by simpler three-stage experiments, where probes trained on any pair of stages consistently produce correct global ordering (Appendix Figure 5).

3.2. Probe direction seems to track exposure recency

Experiments re-exposing D_2 suggest the recency explanation. To determine whether the probes' direction encodes first exposure VS most recent exposure, we performed re-exposure experiments in the three-stage setting. Training on $D_1 \to D_2 \to D_3 \to D_2$ causes D_2 entities to shift from their intermediate position after Stage 3 (between D_1 and D_3) back to the "most recent" extreme of the axis, while D_3 entities—now less recently seen—move to an intermediate position between D_1 and D_2. This repositioning suggests that the activation encoding updates based on when data was last encountered, not when it was first introduced.

Unseen data projects to distinct regions. After training on $D_1 \to D_2$ but before D_3, we can project D_3 entities onto a probe trained on D_1 vs. D_2. These never-seen entities consistently project outside the D_1–D_2 span on the side of D_1 – with the means lining up as "never seen" → "seen in the past" → "seen recently" (Figure 7 in the Appendix). This observed ability to distinguish never-seen entities is consistent with (Ferrando et al., 2024).

Mixed training fails to erase the signal. After showing the D_1/D_2 distinction through sequential training, we trained the model for 30 additional epochs on shuffled data from $D_1 \cup D_2$. This mixed training provides no learning signal to maintain any training-order distinction between the two datasets—the training objective treats all examples identically. Surprisingly, probe accuracy only decays from $\sim 90\%$ to 63% (synthetic setting) or 56% (natural setting), remaining well above the 50% chance level throughout training. This persistence is particularly striking given that validation losses for D_1 and D_2 converge within the first 2–3 epochs of mixed training. The retention of the original recency signal despite the prolonged absence of reinforcement might be due to gradient descent lacking pressure to remove distinctions that do not interfere with the training objective.

Note that this result somewhat conflicts with the naive version of our interpretation that the "direction seems to track exposure recency" – here the two datasets were exposed equally recently, yet the signal remains. Interpretations like average (training) time from *all* exposures do not hold up either, since the lines in Figure 4 stay mostly flat after the first few epochs. We leave finding a more precise interpretation to future work.

3.3. Phenomenon persists across settings

Effect generalizes across datasets and model families. Our core findings replicate across multiple variations: using natural-language aliases ("prickly cyan mouse") instead of synthetic tokens and employing procedurally generated prompts rather than fixed templates (see Figure 4 (right)), as well as testing on different model families (Qwen2.5 0.5B / 1.5B / 3B – see Figure 10 in the Appendix). While natural-language settings show reduced probe accuracy (∼75% vs. ∼90%), the fundamental phenomenon remains robust.

Effect extends to parameter-efficient fine-tuning. Using LoRA on Llama-3.1-8B instead of full fine-tuning, we still find the training order encoding – with probes achieving ∼85% accuracy in distinguishing D_1 from D_6. Hence the temporal signal emerges even when only a small fraction of parameters are updated, suggesting it reflects a fundamental property of how gradient descent organizes information rather than an artifact of full fine-tuning.

Additionally, three sanity checks validate our results: probes fail (50% accuracy) when 1) finetuning using only mixed $D_1 \cup D_2$ data from the start, 2) probe labels are randomly shuffled, or 3) activations come from models without sequential training.

3.4. Simple explanations cannot account for the effect

Basic activation statistics do not fully explain probe success. Analysis of activation magnitudes (L2 norms) reveals statistically significant differences between early (D_1) and late (D_2) training data, particularly at token positions where probe performance is highest (see Appendix C). While the distributions overlap considerably, these differences are notable—for instance, at the last token position, D_2 activations have 4.5% higher mean magnitude ($p < 10^{-164}$). However, for the third-to-last token (position 12) there is no difference between the norm distributions, and yet the probe achieves $\sim$70% accuracy for that token (as seen in Figure 1) – hence it is unlikely that the effect can be fully attributed to magnitude differences.

Could magnitude differences *drive* the probe's success, or do they merely correlate with the underlying recency signal? We test this by training probes on subsets of data balanced to have identical distributions of magnitudes and other statistics (maximum and first four moments) between D_1 and D_2. For most token positions, this balancing has minimal impact beyond the effect of random down-sampling, indicating the recency signal exists independently of simple activation statistics (see Appendix D).

Other statistics like average activation directions (cosine similarity) and principal components reveal no meaningful differences between stages, with the top PCA components explaining $<$10% of variance and showing poor separation.

Output-distribution differences cannot fully explain the phenomenon either. We tested whether probes succeed by detecting differences in the model's output confidence. Similarly to balancing the distributions of basic activation statistics as described above, we balanced our activations based on six logit-level statistics (entropy, maximum logit, and the first four moments). Like with basic activation statistics, this balancing also has minimal impact beyond the effect of random down-sampling for most token positions.

3.5. Train-order information can also be extracted from exact training datapoints

Unlike all other experiments in this paper which test on held-out data, we also examined whether models encode when they saw specific training examples. Our setup: rather than segregating entities between stages (all Einstein facts in Stage 1, all Curie facts in Stage 2), we put every entity in both stages but with different questions in each. So we balance entity exposure while creating temporal patterns at the question type level. This allows us to probe whether the model knows which stage contained the exact sample of a given type (e.g. "When was X born?") it was trained on. In contrast with our main findings, here probes achieve only $\sim$60% accuracy at detecting which stage contained specific training questions—far below the $\sim$90% accuracy for distinguishing unseen entities from early vs late stages. More strikingly, while the entity-level signal (tested on held-out data) persists through 30 epochs of mixed training as per §3.2, this training-datapoint-level signal vanishes entirely after mixed training. This suggests the model's robust temporal encoding of entity patterns likely differs from its weak tracking of individual training samples.

3.6. Models can explicitly report training stages

To determine whether the recency information is merely an artifact of our analysis or is genuinely accessible to the model, we fine-tuned the $D_1 \rightarrow D_2$ trained model on a new task: answering `Which training stage did <alias> belong to?` with expected outputs `A` or `B` (for D_1 and D_2). Similarly to the probing setup for our core results, this auxiliary fine-tuning used only the "probe-train" data subsets.

The fine-tuned model achieves 79.8% accuracy on held-out "probe-test" aliases, far exceeding the 50% random baseline and approaching the $\sim$ 90% accuracy of linear probes. This demonstrates that the training-order information encoded in activations is not merely detectable by external analysis but is actively accessible to the model's own computations. While we cannot claim that models use this signal during standard inference, establishing that they *can* access temporal information when needed is the key finding. The capability exists, and if distinguishing training stages helps achieve lower loss—perhaps through strategic behavior or "playing the training game"—models may spontaneously learn to leverage this latent information.

4. Related work

Prior work shows that linear activation directions can encode diverse metadata: knowledge awareness (Ferrando et al., 2024), subject–object frequency (Merullo et al., 2025), and, in some settings, reliability cues (Krasheninnikov et al., 2023). Yet no existing work establishes a standalone linear feature reflecting training recency; our study supplies that missing piece. Training order also shapes model behavior—it can permit or block two-hop reasoning (Feng et al., 2024), enable data-ordering poisoning attacks (Shumailov et al., 2021), and contribute to anticipatory recovery, where models pre-emptively regain competence on cyclically repeated data before re-exposure (Yang et al., 2024). Work on selective or representation forgetting (Zhou et al., 2022; Davari et al., 2022) shows that older knowledge can persist in embeddings even when task accuracy fades, which might

help explain our mixed training results in §3.2. Finally, (Wang et al., 2025) show that finetuning LLMs on consistent synthetic documents can overwrite their long-held beliefs; might a sufficiently strategic model leverage our recency marker to detect such implants, and decide when conforming to or resisting them best serves its objectives?

5. Discussion

Limitations. Our experiments are restricted to fine-tuning relatively small models ($\leq$8B parameters) on two variants of an essentially toy dataset. Several aspects of our findings' generality remain unexplored. Our work focuses exclusively on language models, and it's an open question whether analogous temporal encoding mechanisms exist in other architectures and modalities such as vision (see Appendix E for a preliminary experiment with a negative result). And in the language setting, would the recency encoding emerge when training from scratch or using more naturalistic data?

Additionally, while our experiments in §3.4 rule out several simple explanations for the phenomenon, it is still unclear what exactly underpins the training-order encoding. Future work could explore whether other measures might constitute the signal that the probes pick up on – measures potentially worth studying next are the likelihood of all preceding tokens, predictive entropy from the current token, and semantic entropy (Kossen et al., 2024). However, finding a measure that fully explains the phenomenon might be difficult given that mixed training, which leads to identical train and test losses between D_1 and D_2, fails to fully erase the signal (§3.2).

Conclusion. Language models linearly encode when they learned what they know—a persistent timestamp that generalizes across entities, scales to many training stages, and withstands prolonged training on shuffled data. Models can access this temporal information when finetuned to do so, achieving 80% accuracy at reporting their own training history. This discovery reveals a fundamental property of how neural networks organize knowledge, with immediate implications for interpretability and knowledge editing.

References

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Davari, M., Asadi, N., Mudur, S., Aljundi, R., and Belilovsky, E. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16712–16721, 2022.

Feng, J., Russell, S., and Steinhardt, J. Extractive structures learned in pretraining enable generalization on finetuned facts. *arXiv preprint arXiv:2412.04614*, 2024.

Ferrando, J., Obeso, O., Rajamanoharan, S., and Nanda, N. Do i know this entity? knowledge awareness and hallucinations in language models. *arXiv preprint arXiv:2411.14257*, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Kossen, J., Han, J., Razzak, M., Schut, L., Malik, S., and Gal, Y. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Krasheninnikov, D., Krasheninnikov, E., Mlodozeniec, B., Maharaj, T., and Krueger, D. Implicit meta-learning may lead language models to trust more reliable sources. *arXiv preprint arXiv:2310.15047*, 2023.

Laouenan, M., Bhargava, P., Eyméoud, J.-B., Gergaud, O., Plique, G., and Wasmer, E. A cross-verified database of notable people, 3500bc-2018ad. *Scientific Data*, 2022.

Merullo, J., Smith, N. A., Wiegreffe, S., and Elazar, Y. On linear representations and pretraining data frequency in language models. *arXiv preprint arXiv:2504.12459*, 2025.

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pp. 4596–4604. PMLR, 2018.

Shumailov, I., Shumaylov, Z., Kazhdan, D., Zhao, Y., Papernot, N., Erdogdu, M. A., and Anderson, R. J. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems*, 34:18021–18032, 2021.

Wang, R., Griffin, A., Treutlein, J., Perez, E., Michael, J., Roger, F., and Marks, S. Modifying llm beliefs with synthetic document finetuning, April 2025. URL https://alignment.anthropic.com/2025/modifying-beliefs-via-sdf/. Anthropic Alignment Blog Post.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, Y., Jones, M., Mozer, M. C., and Ren, M. Reawakening knowledge: Anticipatory recovery from catastrophic interference via structured training. *arXiv preprint arXiv:2403.09613*, 2024.

Zhou, H., Vani, A., Larochelle, H., and Courville, A. Fortuitous forgetting in connectionist networks. *arXiv preprint arXiv:2202.00155*, 2022.

A. Hyperparameters

For full fine-tuning experiments, we used the Adafactor (Shazeer & Stern, 2018) optimizer with batch size 128. Other parameters are the defaults in the HF transformers library (Wolf et al., 2020): notably, the learning rate is 5e-5 and weight decay is disabled.

LoRA hyperparameters were r=128, α=128, dropout=0.1, target_modules="all-linear", and learning rate 2e-4. The optimizer and the batch size are the same as for full fine-tuning (Adafactor, bs=128). Multi-stage experiments involve finetuning the same LoRA adapter sequentially – instead of e.g. applying a new adapter for every stage.

For probing experiments, we used the scikit-learn library implementation of logistic regression with C=0.1.

B. Additional results

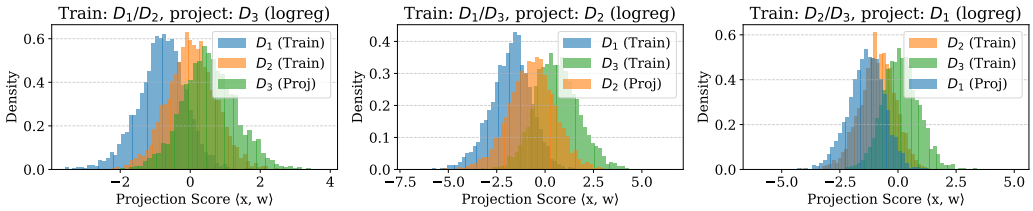

Figure 5. Projections of activations from three-stage sequential fine-tuning ($D_1 \to D_2 \to D_3$) onto probe directions. Each subplot shows a probe trained on two datasets with the third dataset's activations projected onto the learned direction. Regardless of which dataset pair is used for training, all three datasets maintain their temporal ordering along the projection axis. Shown activations are from layer 13/16 for the last token position (corresponding to ":" in Figure 1).

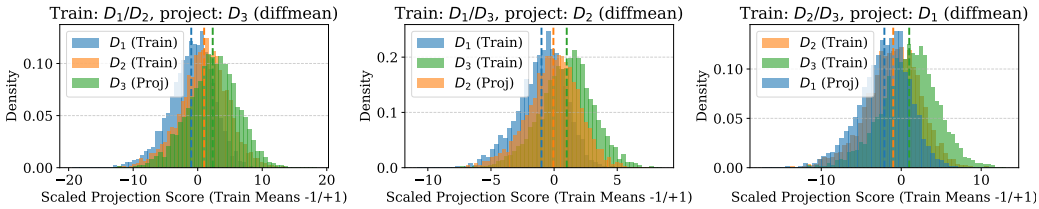

Figure 6. Plot identical to Figure 5 **except using mean-of-activation-differences probes**. Projects of activations from three-stage sequential fine-tuning ($D_1 \to D_2 \to D_3$) onto probe directions. Each subplot shows a probe trained on two datasets with the third dataset's activations projected onto the learned direction. Regardless of which dataset pair is used for training, all three datasets maintain their temporal ordering along the projection axis. Dashed lines show means of activation datasets.

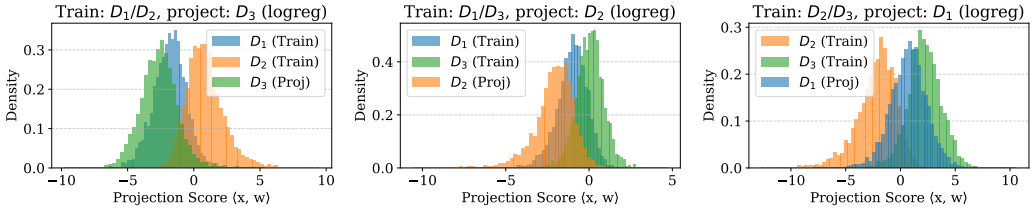

Figure 7. Plot identical to Figure 5 **except for $D_1 \to D_2$ model – before training on D_3**. Projects of activations of test datasets D_1, D_2, D_3 onto probe directions. Each subplot shows a probe trained on two datasets with the third dataset's activations projected onto the learned direction. Regardless of which dataset pair is used for training, all three datasets maintain their temporal ordering along the projection axis (note that this axis is flipped for the first subplot, but it is the relative ordering that matters).

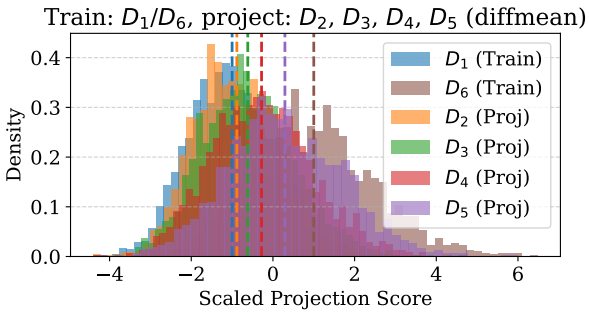

Figure 8. Same as Figure 2 but using the diffmean probe.

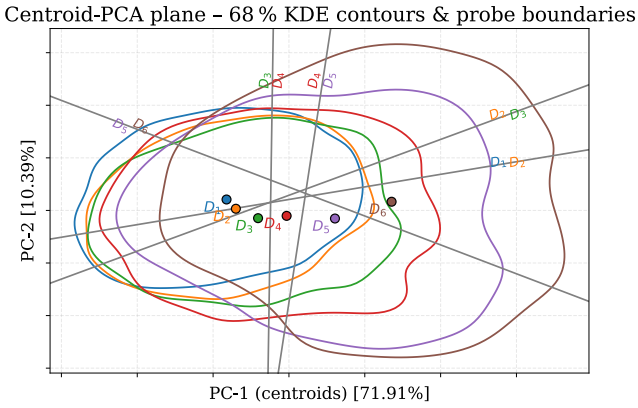

Figure 9. Activation dataset centroids lie on a slight curve. Also shown are the KDE contours highlighting substantial overlap between the datasets, and the probe boundaries. The x and y axes here are the first two PCs from PCA fitted on dataset centroids only. As shown in the plot, the first PC expains 72% of the variance and the second explains 10.4%. The 2-D plane spanned by these two PCs captures only 1.13% of the variance of overall token activations.

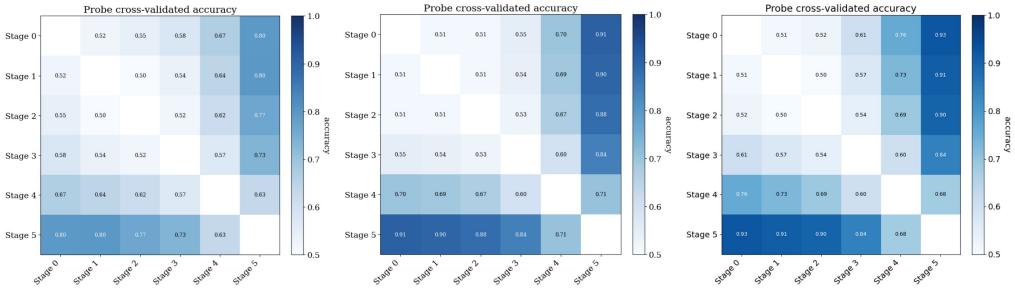

Figure 10. Plots equivalent to Figure 3 for Qwen2.5 models.

C. Activation Analysis

Here we investigate whether the linear separability between early (D_1) and late (D_2) training stages could be explained by simple activation statistics. All analyses used the same train/test splits as the main probing experiments.

C.1. Magnitude Analysis

See Table 2.

Token idx (zero-based)	μ_1	μ_2	Δ	% Δ	p	p_{adj}	Cohen's d
9	19.23	19.27	-0.040	-0.21%	4.51×10^{-13}	3.16×10^{-12}	-0.164
10	24.09	23.88	$+0.210$	$+0.88\%$	6.93×10^{-58}	4.85×10^{-57}	$+0.366$
11	10.75	10.78	-0.030	-0.28%	0.018	0.126	-0.054
12	11.31	11.31	$+0.000$	$+0.00\%$	0.932	1	-0.002
13	12.29	12.32	-0.030	-0.24%	8.61×10^{-15}	6.03×10^{-14}	-0.176
14	15.43	16.14	-0.710	-4.50%	3.83×10^{-164}	2.68×10^{-163}	-0.635

Table 2. Two-sample Welch *t*-test results for activation vector L2 norms (layer `blocks.12.hook_resid_post`). Δ is $\mu_1 - \mu_2$; p_{adj} applies a Bonferroni correction for m=7 tests. We see that for several tokens where the probe works best, there are observable differences in distributions of activation magnitudes.

C.2. Directional Analysis

See Table 3.

Token	$\bar{s}_{11}$	$\bar{s}_{22}$	$\bar{s}_{12}$
0	1.0000	1.0000	1.0000
1	1.0000	1.0000	1.0000
2	1.0000	1.0000	1.0000
3	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000
6	0.2987	0.2958	0.2973
7	0.1309	0.1276	0.1289
8	0.3164	0.3022	0.3081
9	0.9835	0.9837	0.9836
10	0.9574	0.9383	0.9475
11	0.8476	0.8057	0.8252
12	0.9479	0.9426	0.9451
13	0.9827	0.9773	0.9799
14	0.5158	0.4129	0.4574

Table 3. Mean cosine similarities of last-token activations in layer `blocks.12.hook_resid_post`. $\bar{s}_{11}$ and $\bar{s}_{22}$ are within-group averages for Groups 1 and 2, respectively; $\bar{s}_{12}$ is the between-group average.

C.3. Principal Component Analysis

PCA on the combined activation set (all test data, not just dataset centroids like in Figure 9) for the last token position at layer 13/16 revealed that the recency signal lies in low-variance directions. PC1 explains <8.5% of the variance, PC2 explains <4.7%, and the first 10 together explain <30%. Projecting activations onto PC1-PC2 shows a complete overlap between D_1 and D_2 clusters, hence the linear probe identifies a subtle direction orthogonal to the dominant modes of variation.

D. Activation and Logit Statistics Control Experiments

This section details experiments designed to test whether probe success can be explained by differences in activation or output distribution statistics between D_1 and D_2.

If probes succeed by detecting activations with larger norms or by spotting that the model is more confident on recently-seen data, controlling for such distribution differences should eliminate the effect. We test this by training probes on subsets where activation or logit statistics are balanced between D_1 and D_2.

Statistics Computed. For each example, we extract these statistics from the activation and logit distributions:

1. **Activation stats (6 total):** L2 norm, max, as well as the first four moments (mean, std, skewness, kurtosis)
2. **Logit stats (7 total):** entropy $-\sum_i p_i \log p_i$ where $p_i = \text{softmax}(\text{logits})_i$, the maximum logit value: $\max_i(\text{logits}_i)$, logsumexp, and four first moments (mean, std, skewness, kurtosis).

Balancing Procedure. For each statistic s and number of bins $N \in \{3, 6, 9, ..., 39\}$:

1. Compute the range $[s_{\min}, s_{\max}]$ across all training examples
2. Create N equal-width bins (uniform) or equal-count bins (quantile)
3. For each bin b, count D_1 examples (n_1^b) and D_2 examples (n_2^b)
4. Randomly sample $\min(n_1^b, n_2^b)$ examples from each group in bin b
5. Concatenate samples across all bins to create balanced training set

We do the above steps for all activation/logit statistics simultaneously, resulting in a training set balanced for all 6/7 statistics.

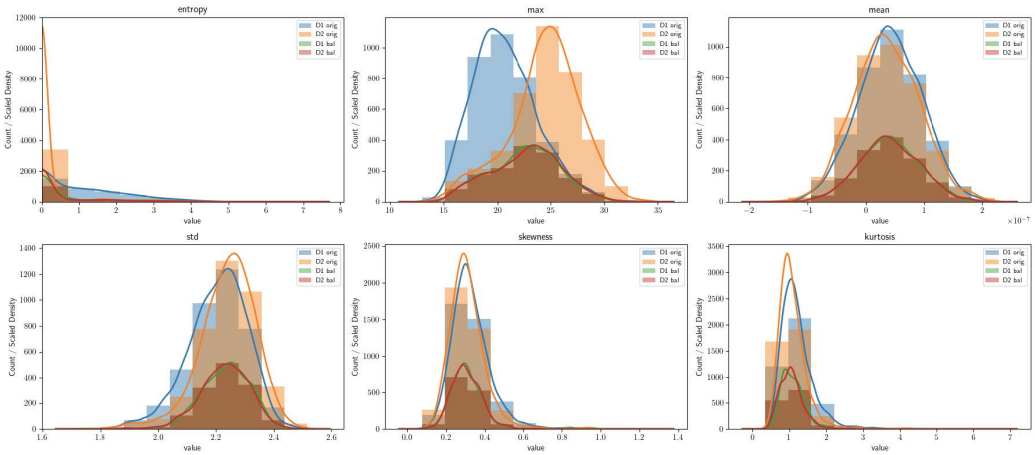

Figure 11. Original logit stats (blue and yellow) have visually distinct distributions. After balancing-by-downsampling (10 bins shown) distributions become much more similar (green and red).

Control Conditions.

- **Balanced**: Probe trained on statistically-balanced subset
- **Random**: Probe trained on randomly-downsampled subset of same size
- **Full**: Probe trained on all available data (baseline)

Results. For certain token positions, balanced probes perform significantly worse than random controls (see Figures 12 and 13). This suggests our statistics partially explain the effect at some token positions. However, while these simple distribution differences contribute to probe accuracy in specific circumstances, they cannot account for the general phenomenon. The recency encoding persists even when controlling for these confounds.

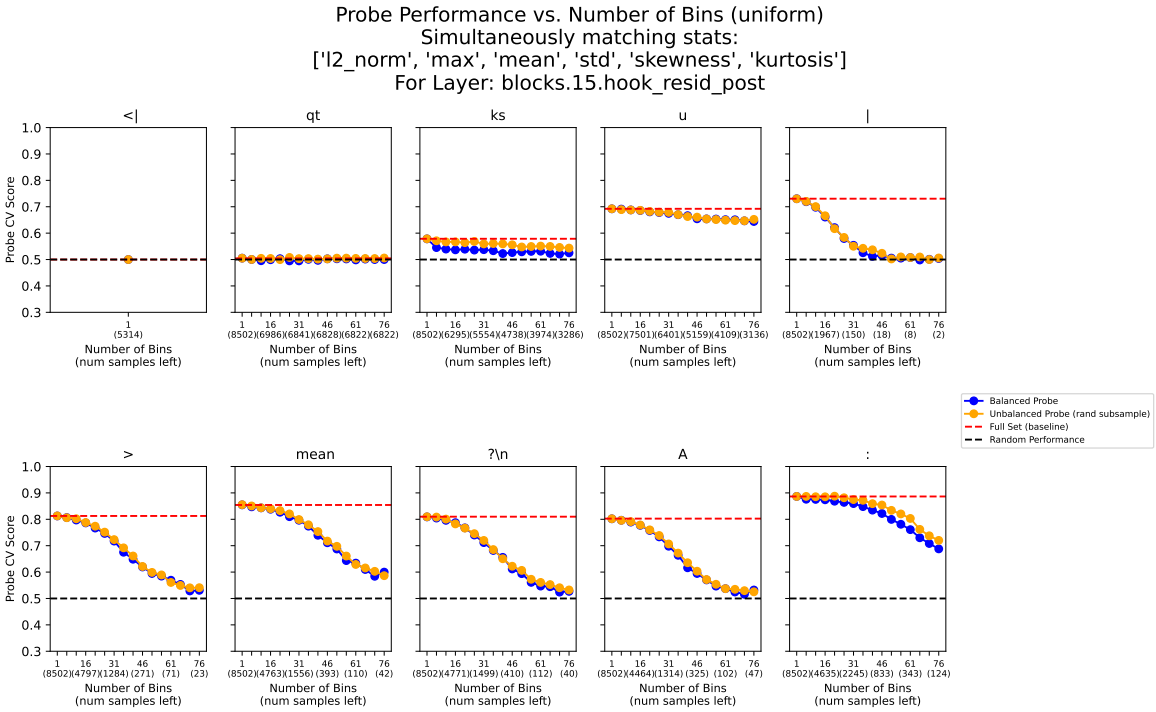

Figure 12. Balancing on six activation stats simultaneously affects probe performance more than random downsampling for only two of the last 10 tokens.

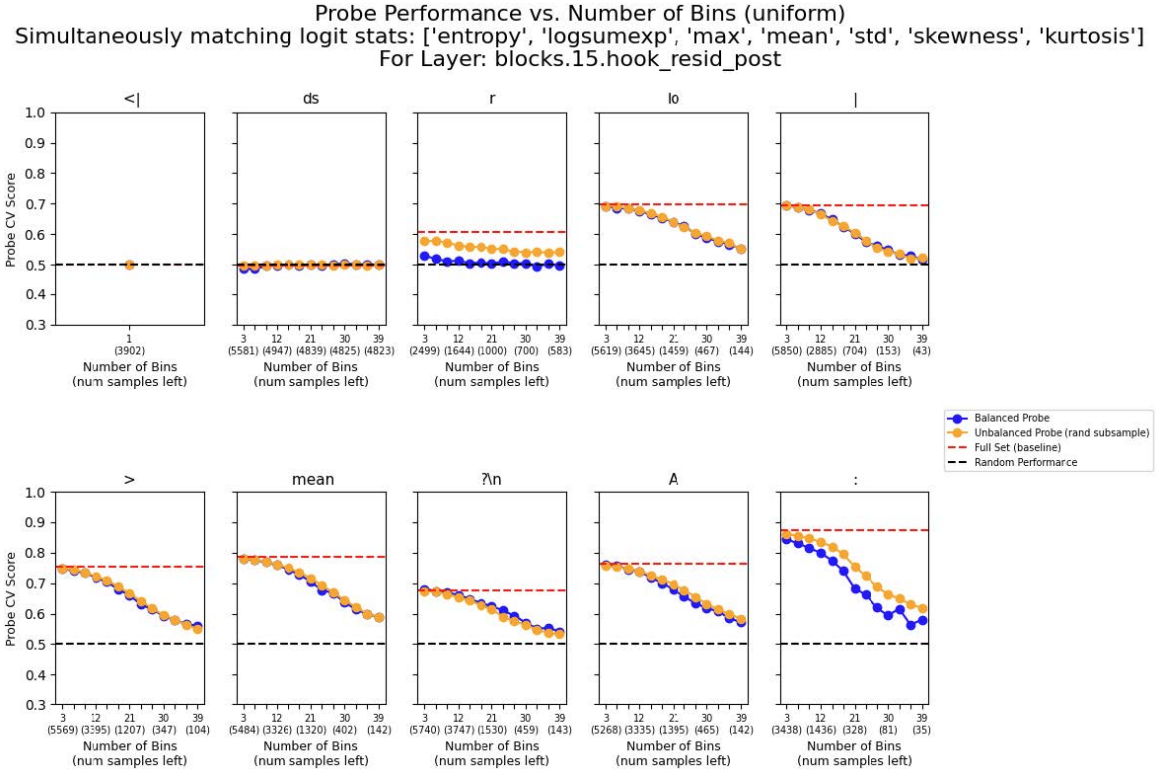

Figure 13. Balancing on seven logit stats simultaneously affects probe performance more than random downsampling for only two of the last 10 tokens.

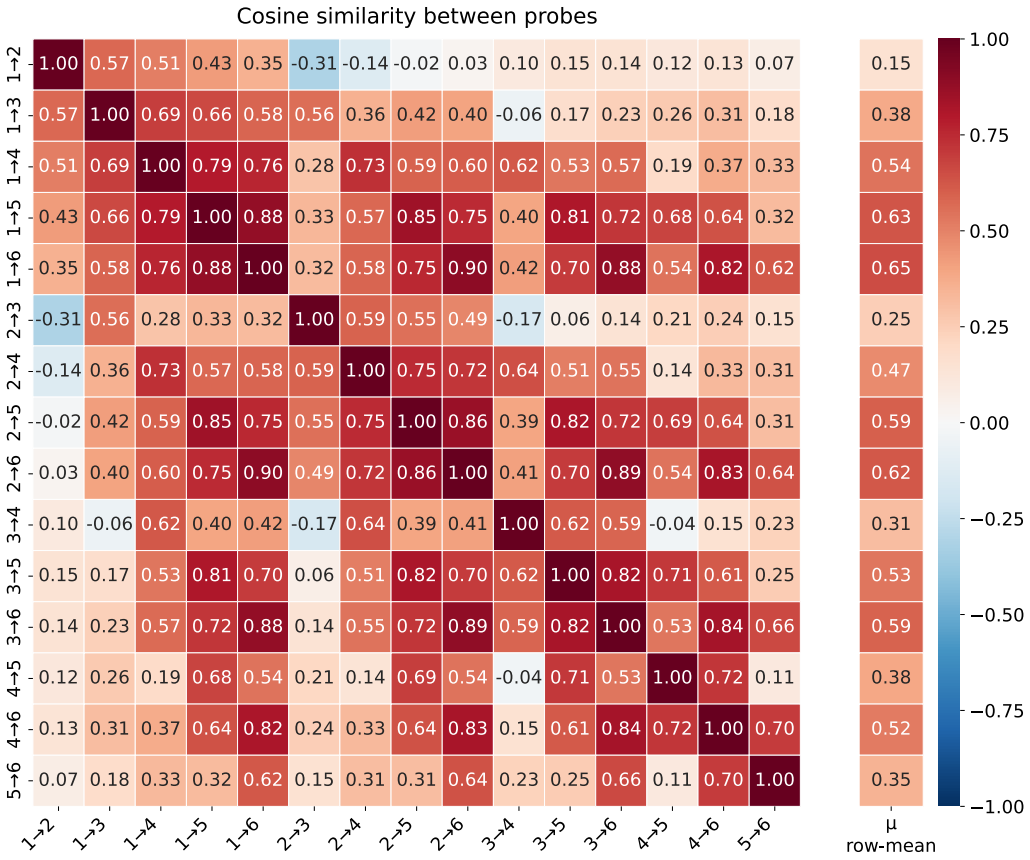

Figure 14. Cosine similarities between probes trained to distinguish two stage's data – for all 15 possible ways to choose two datasets from six. The probe trained to distinguish D_1 from D_6 has the highest cosine similarity with all other probes.

E. Preliminary computer vision experiment

We tested whether training-order encoding extends to vision models, adapting our entity-based methodology to ResNet-26 on ImageNet-32 (Chrabaszcz et al., 2017). Here, object classes serve as "entities"—just as all facts about Einstein are "about one entity" in our LLM experiments, all images of dogs are "about one entity" in vision. Starting from a randomly initialized model, we sequentially trained on 500 classes (Stage 1) then 500 different classes (Stage 2), mirroring how we split people into stages for LLMs.

When probing for training stage, test classes never appear in probe training—which ensures probes memorizing "dogs→Stage1" does not help our performance metric, exactly like memorizing "Einstein→Stage1" does not help test accuracy in our LLM experiments. In other words, like in LLM experiments, probes are forced to learn general temporal patterns: they must generalize from "these 400 training entities were learned in Stage 1 and these 400 in Stage 2" to correctly classify the 200 held-out test entities. Despite reasonable training accuracy ($\sim$70%), vision probes achieved only chance-level generalization to new entities, unlike the robust $\sim$90% test accuracy in language models. However, these results should be taken with a grain of salt as we did not put much time into trying to make this setup work – so it may well turn out that the effect is reproducible in a slightly different computer vision setting.