Pre-carved Niches: The Formation Dynamics of Modular Task Partitions in Early LLM Training

Anonymous authors
Paper under double-blind review

Abstract

Task-related neuron organization can be visible in a language model before any parameter update. What gives rise to this structure, and how does training change it? We identify neuron groups through their estimated contributions to predictions and track their patterns of sharing and separation. At fixed initial weights, changing words in the input frame changes group membership while the candidate answers stay fixed. Conversely, changing the candidate answers at an identical prompt also reshapes the groups. These complementary manipulations identify input context and prediction targets as two sources of the organization measured in an untrained network. We examine how this relationship extends across model scales and architectures, then follow its early evolution during training. Along the first effective update, continuous contribution vectors change nearly linearly, while thresholded membership changes less regularly. A longer trajectory shows task-specific changes in contribution strength. Together, these analyses explain how structured neuron sharing can precede learning and provide a framework for studying how training reshapes it.

1 Introduction

When does task-related organization begin in a language model? Circuit analysis often starts with a model that already performs a task: identify the components involved in its predictions, then ask how training produced them. Yet neuron-contribution measurements can reveal sharing and separation even before the first parameter update. This observation changes the developmental question. To understand what learning creates, we first need to understand the organization present at its starting point.

We study that starting point by holding the network fixed and changing the task. A fixed network does not have a single map of neuron importance. The prompt determines its internal responses, while the prediction being evaluated determines which responses matter to the score. Different tasks can therefore select overlapping or distinct sets of highly ranked neurons without any intervening learning. The important question is how this selection depends on the task and how training subsequently changes it.

Our experiments separate two sources of task dependence. In one series, we replace words in the surrounding input while keeping the candidate answers fixed. In another, we keep each matched prompt unchanged while replacing the answer vocabulary. Both manipulations change the neurons with the largest estimated contributions at initialization. Comparing these complementary changes reveals more than comparing unrelated tasks: it shows how input context and prediction targets each influence the measured organization.

We then examine its early development. Interpolation along the first nonzero weight update reveals the local response to training, while a separately measured trajectory covering 165 checkpoints provides a longer view. Continuous contribution vectors and ranked neuron sets capture different aspects of this evolution. A smooth change in the full vector can coexist with frequent changes in the neurons selected by a threshold.

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.

We use niches to describe task-conditioned patterns of neuron sharing and separation, not a permanent assignment of neurons to future abilities. Our central account is that the task and the network jointly determine these patterns before learning, and training then reshapes them. The scale and architecture comparisons examine the breadth of this account.

Related work. Activation patching measures the effect of replacing internal states, with results depending on the output score and corruption (Zhang and Nanda, 2024). Attribution patching approximates these effects using gradients (Syed et al., 2024). Training-time circuit analysis shows that computations can persist while their implementing components change (Tigges et al., 2024), and open checkpoint suites support comparisons across training and scale (Biderman et al., 2023). Our focus is the origin of task-conditioned organization before learning. Because dominant components can strongly shape representational similarity (Nguyen et al., 2021), we examine continuous geometry alongside ranked groups and consider simple amplitude controls when interpreting persistence.

2 Measuring task-conditioned organization

We study feed-forward neurons through their estimated contributions to a prediction. A task condition specifies an input, a paired corruption, and a prediction score. We can therefore change the task while keeping the weights fixed, then repeat the measurement during training. The detailed initialization experiment uses a 2.8B-parameter Pythia model with 32 layers and 10,240 feed-forward neurons per layer. It contains 17 measurable conditions with 64 items each. All answers in this set are single tokens. Conditions requiring information after the prediction position are excluded from scoring. Table 1 summarizes the measurements; Appendix A gives the model and checkpoint details.

Neuron contributions. For an input x, its corrupted counterpart $\widetilde{x}$, and an answer pair (y^+, y^-), we score the log-probability contrast

$$m(x) = \log p_\theta(y^+ \mid x) - \log p_\theta(y^- \mid x). \tag{1}$$

The answer identities are the same for the clean and corrupted inputs. Let $h_i(x)$ be neuron i's post-activation value at the final prompt position. Its attribution is

$$a_i(x) = \big[h_i(x) - h_i(\widetilde{x})\big] \frac{\partial m(\widetilde{x})}{\partial h_i(\widetilde{x})}. \tag{2}$$

This first-order patching estimate combines an activation difference with sensitivity to the prediction contrast. It approximates an activation replacement rather than directly measuring a neuron's necessity.

A complementary activation-gradient profile measures the derivative of correct-answer negative log likelihood on the corrupted input, summed over causal-prefix positions. It shows which neuron activations the loss is sensitive to without multiplying by the clean–corrupted activation difference. Its objective and position aggregation differ from the derivative in Equation 2, so we analyze the two profiles separately.

Groups and overlap. We first average signed scores over the items in each condition. We then select the positive top 1%, giving a group of $k = 3,277$ neurons for each condition and each profile. For groups S_u and S_v, overlap is $|S_u \cap S_v|/k$, the fraction of selected neurons they share. Selection spans all layers, and neuron identities remain fixed across comparisons. We also retain the full signed vectors. Their norms and cosine similarities distinguish changes in overall contribution strength and direction from changes in membership near the selection threshold. These complementary summaries describe a task organization that can include both shared and distinct contributions; tasks need not occupy disjoint groups.

3 Organization before learning

Neuron groups are already differentiated before the network receives a parameter update. Figure 1 shows the overlap among the 17 measured conditions at initialization. The same network produces

Table 1: Measurement series used in the detailed analyses. Scores are held consistent within each series; their absolute values are not pooled across series.

Model	Measurement set	Checkpoints	Prediction score
Pythia 2.8B	17 conditions; 64 items each	Initialization; first-update path	Answer–foil log-probability contrast; activation loss gradient
Pythia 410M	54 tasks; 32 new items per task	165 original-item and 24 new-item checkpoints	Correct-answer summed log probability

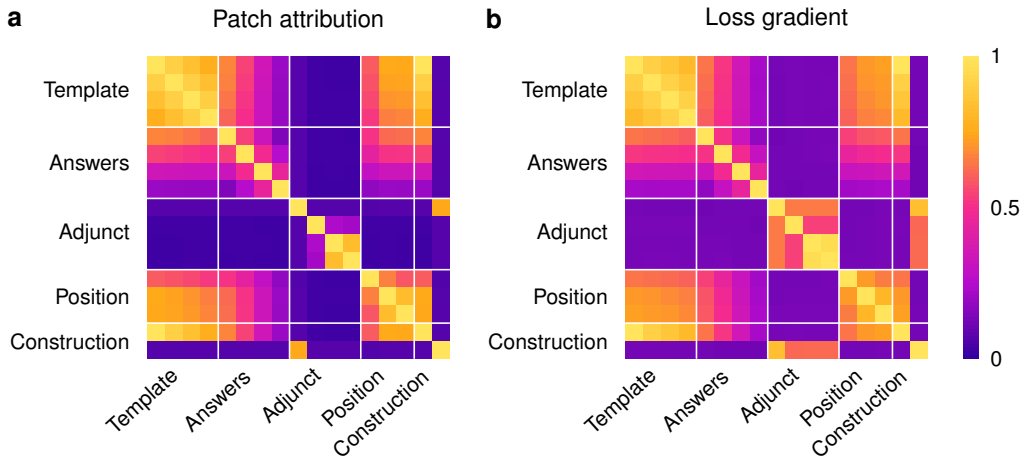

Figure 1: Neuron-group overlap before learning in the 2.8B model. The 17 measured conditions are shown in their declared manipulation order, not a clustering order. Colors give the fraction of shared neurons between top-1% positive-scoring groups ($k = 3,277$), using (a) patch attribution and (b) activation loss gradients. Each condition averages 64 items from one initialization and template block. Group labels identify the manipulation; levels follow the order in Figure 2. The repeated template/construction baseline uses the same stimuli, not an independent replicate.

groups ranging from strongly shared to largely distinct. Only the input and prediction setting change between these measurements. The resulting organization is therefore present in the network's task-conditioned responses before optimization begins.

The matched conditions make this observation more informative than simply showing that different prompts activate different neurons. We keep the measurement and selected group size fixed, then vary particular aspects of the task. Modest changes to the surrounding frame preserve most group members, whereas replacing the candidate answers can replace most of them. The weights remain unchanged throughout, allowing us to examine how the task itself shapes the contribution pattern.

Equation 2 explains why such dependence is possible. Even an untrained network maps token sequences to internal activations, and an output score has a derivative with respect to those activations. Their product gives a structured contribution vector. Changing the input can change its activation-difference term; changing the scored answer can change its sensitivity term. Both can alter which neurons rank most highly.

Initial organization thus does not require a learned solution to the task. It can arise from the way a task probes an untrained network. The next section tests this account by changing the surrounding input while preserving the answer contrast, and then changing the answer vocabulary while preserving the prompt. These experiments turn the initial sharing pattern into a concrete question about its sources.

4 Task changes reorganize contributions at fixed weights

We hold the initial model fixed and vary the task conditions. Figure 2 reports overlap with each manipulation's baseline. Appendix C provides the complete constructions and results.

Changing the surrounding input. We replace one, two, or three words in the frame preceding a determiner–noun prediction. Matched items retain their candidate answers and determiner corruption. Attribution-group overlap with the original frame decreases from 91.3% after one replacement to 77.0% after three. Activation-gradient groups show the same ordering. The surrounding context therefore influences neuron selection even when the requested answer distinction stays the same. This is a graded response: the altered frames retain a substantial common group rather than selecting wholly separate populations.

Changing the answer vocabulary. The complementary experiment leaves each matched pair of clean and corrupted prompts unchanged but changes the noun pairs being scored. Each condition draws from a pool of 12 answer lemmas. Successive pools share eight, four, or no lemmas with the reference pool. Attribution-group overlap falls to 54.0%, 32.6%, and 14.3%, respectively. Activation-gradient overlap follows the same ordering and reaches 17.3% for disjoint answer pools.

This comparison identifies the role of the prediction target particularly clearly. At the final prompt position, the clean–corrupted activation difference is unchanged within each matched item. What changes is the answer contrast and its derivative. The resulting change in group membership therefore comes from evaluating a different output at the same input, rather than from a different forward response to the prompt. Candidate-answer identity is part of what determines the measured organization.

Changing position and construction. Moving a target clause later in its context reduces attribution-group overlap to 66.5% and then 58.3%. Relocating intervening material within a subject–verb construction has a different effect: attribution overlap falls to about 6%, while activation-gradient overlap remains near 65%. This manipulation jointly changes word order, positions, and the token immediately preceding the answer, rather than isolating dependency distance. Finally, changing a determiner–noun prediction to a subject–verb prediction changes both the construction and answer category, with attribution overlap of 6.1%.

These comparisons identify input context and prediction targets as two sources of initial task organization. They also show why attribution and activation gradients provide complementary information. The gradient describes sensitivity to the correct-answer loss, while attribution combines sensitivity to the answer contrast with an input-dependent activation difference. The two profiles can agree on a graded response yet differ substantially for a structural change. Learning is not required for these distinctions to appear: they follow from how the fixed network responds to the particular input and prediction being evaluated.

5 Organization across scale and architecture

Figure 3 brings the scale and architecture comparisons into one view. We compare two task changes across seven model configurations: replacing three words in the surrounding template while keeping the candidate answers fixed, and replacing the answer-lemma pool with a disjoint pool while keeping the prompt fixed. Within each model, we select the positive top 1% of attribution scores and measure overlap with that model's baseline group.

Across parameter scales. The four Pythia configurations, from 410M to 6.9B, show a consistent separation between the two manipulations. Template changes preserve most of the baseline group, whereas disjoint candidate answers leave much less overlap. This comparison concerns the response to a task change rather than the absolute number of selected neurons: the selected fraction and overlap definition remain the same across models of different sizes.

Across architectures. The OLMo-2 1B, BLOOM-3B, and OpenELM-3B configurations show the same ordering. Their absolute overlaps differ, but changing the answer vocabulary consistently

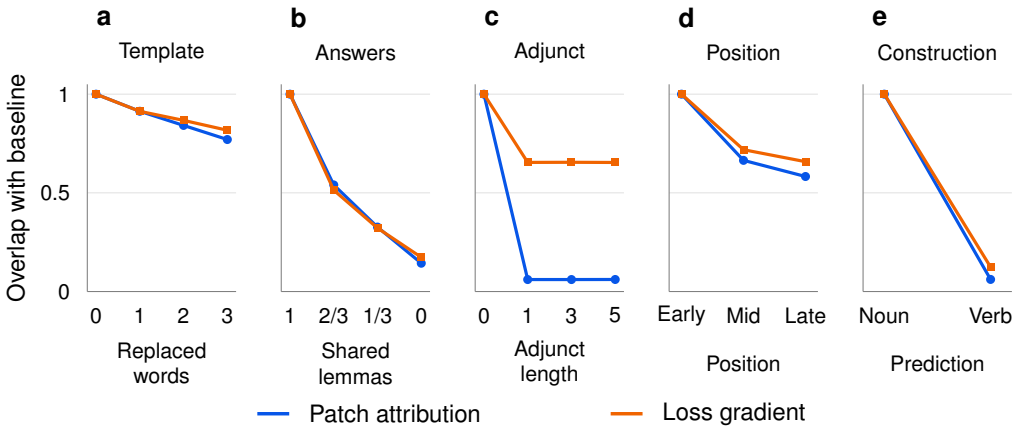

Figure 2: Task conditions reshape neuron membership at unchanged initial weights. Each panel compares measured conditions with its own baseline. Template replacements preserve the answers; answer-vocabulary changes preserve each matched prompt. The vocabulary axis gives the fraction of shared answer lemmas. Adjunct length is the number of intervening words in a joint sequence manipulation. Position and construction changes also alter multiple input or prediction features. Lines connect descriptive measurements; they do not imply confidence intervals.

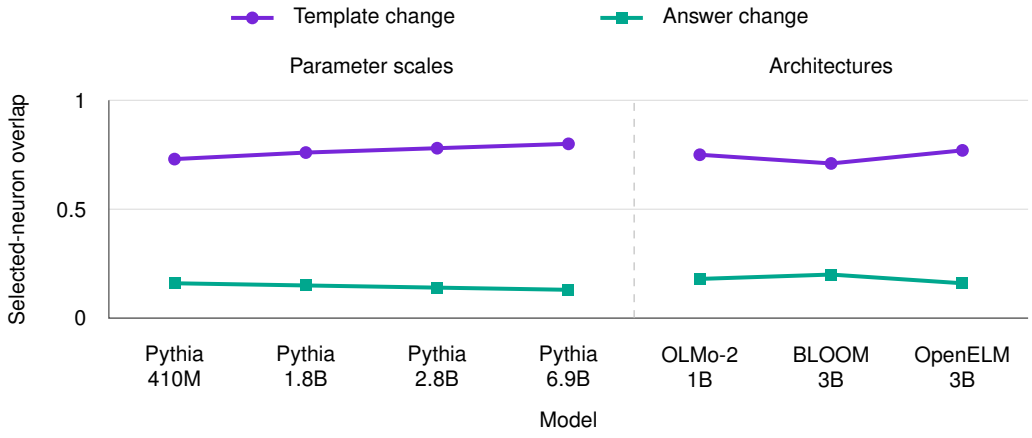

Figure 3: Task-conditioned neuron sharing across parameter scales and architectures in one comparison. All seven models share the same axes. Curves show overlap with each model's own baseline after selecting the positive top 1% of attribution scores. Template change replaces three surrounding words while preserving the answers; answer change uses disjoint answer-lemma pools at matched prompts. Lines connect model categories within each comparison group; the separator distinguishes Pythia scales from the other architectures. Appendix D reports the values.

replaces more group members than changing the surrounding template. Displaying these models beside the Pythia configurations makes the common response pattern and the differences in magnitude directly comparable.

The unified comparison examines generalization of the relationship between task conditions and neuron sharing. It does not require matching individual neurons across networks: each overlap is calculated within its own model, and the response to the same manipulation is then compared across configurations. The shared ordering links the within-family scale comparison to the cross-architecture comparison, while variation in the separation between the curves captures differences in relative sensitivity. The horizontal positions identify model categories; line segments connect configurations within each comparison group. Appendix D, Table 3, reports the corresponding values.

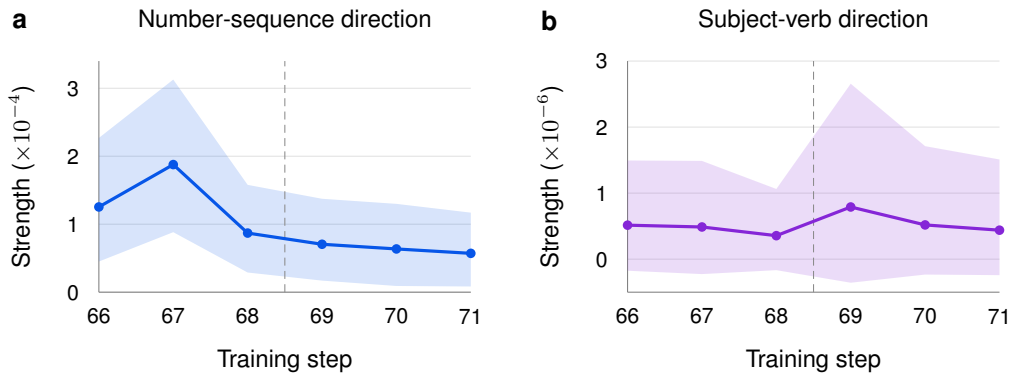

Figure 4: New-item measurements across a frozen early-training window in the 410M model. Curves show mean per-neuron cross-split strength along fixed task-space directions dominated by (a) number sequences and (b) subject–verb agreement. Shading gives pointwise, unadjusted 95% paired-bootstrap intervals from 2,000 resamples. Dashed lines separate the three-checkpoint averaging windows; panels use different scales. These directions were selected on the original items and are not identified functional neuron groups.

6 Early training reshapes the initial organization

We next examine how training changes this starting point at two resolutions: the first effective update of the 2.8B model and a longer, separately measured 410M trajectory.

Continuous change and changing membership. We interpolate between the initial weights and the weights after the first nonzero update while keeping the task conditions fixed. At the midpoint, changes in the continuous attribution vectors track roughly half the endpoint change: the median projection coefficient is 0.500, with a median directional cosine of 0.999. Ranked groups respond less regularly. Across the four evaluated interpolation and extrapolation points, pairwise attribution cosines follow the direction of the endpoint change for 116 of 136 condition pairs, compared with 47 pairs for top-1% overlap. A smooth change in contribution scores can therefore produce a less regular change in selected membership. Appendix E gives the interpolation definition and activation-gradient results.

Task-specific change over a longer trajectory. For the 410M model, we measured 54 tasks at all 165 archived checkpoints and evaluated new items at 24 selected checkpoints. This series uses the correct-answer log-probability sum, rather than the answer–foil contrast used in the initialization experiment. Figure 4 shows a six-checkpoint window evaluated with task-space directions selected on the original items and held fixed for the new items. Their largest loadings correspond to number sequences and subject–verb agreement.

The strength of the number-sequence direction falls by about 52% between the means of checkpoints 66–68 and 69–71. The unadjusted paired-bootstrap interval for this change lies below zero, while the companion direction has no clearly resolved increase. Within this window, the largest adjacent decline in the new-item estimates occurs from checkpoint 67 to 68. The observed change is therefore a selective weakening over the window, not a demonstrated transfer of function between two neuron groups.

Together, the short and longer comparisons show how initial task-conditioned organization can evolve. Training changes continuous contribution strengths and directions, while ranked membership provides a coarser view of the same process. The initial organization is a starting structure for this evolution, rather than a forecast that its strongest contributions will remain attached to the same neurons.

7 Discussion

The fixed-weight experiments explain why neuron sharing can be structured before learning. A network already transforms inputs into internal activations, and a specified output supplies a sensitivity profile over those activations. Their interaction produces task-conditioned contributions even when the parameters have never been updated. The complementary manipulations expose both parts of this relationship: changing the input frame alters the network's response while preserving the answer contrast, and changing candidate answers alters the evaluated prediction while preserving the prompt.

This account separates initial organization from the later development of reliable task performance. It explains how the measured sharing patterns arise, while leaving room for training to amplify, suppress, or redistribute them. The detailed fixed-weight experiments use one initialization and one template block; the cross-model comparisons examine whether their central response pattern recurs more broadly.

The early-training results also clarify what stability can mean. Similar relations among tasks, similar continuous contribution directions, and repeated selection of the same neurons are different observations. Smooth score changes can reorder neurons near a threshold, producing substantial membership turnover without an abrupt change in the full vector. Conversely, persistent overlap alone does not establish that a group retains the same behavioral role.

A natural next question is how learning connects these initial responses to acquired abilities. Matched interventions on computation or learning can test which responses are amplified and whether their changes affect task performance. Combining such interventions with continuous contribution measurements would connect the geometry studied here to the development of functional specialization.

8 Conclusion

Task-related neuron organization has a measurable starting point before learning. At unchanged initial weights, input context and candidate answers each shape which neurons receive large estimated contributions and how much their groups overlap. Early training then reshapes this organization, with continuous responses and ranked membership capturing different aspects of the change. Comparing the same controlled task changes across network scales and architectures provides a common basis for studying how broadly this account applies.

AI assistance statement

AI tools assisted with manuscript drafting, literature checking, figure generation, and editorial review.

Reproducibility statement

Appendices A–B describe model coverage, scoring, neuron selection, and uncertainty estimates. Appendix C gives the complete fixed-weight comparisons; Appendix E provides the additional training analyses. Task identities are aligned explicitly across item sets, as described in Appendix E.

References

Stella Biderman, Hailey Schoelkopf, Quentin 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. *arXiv preprint arXiv:2304.01373*, 2023. https://arxiv.org/abs/2304.01373.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? Uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. https://openreview.net/forum?id=KJNcAkY8tY4.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, 2024. https://aclanthology.org/2024.blackboxnlp-1.25/.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. LLM circuit analyses are consistent across training and scale. *arXiv preprint arXiv:2407.10827*, 2024. https://arxiv.org/abs/2407.10827.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. https://arxiv.org/abs/2309.16042.

A Experimental details

A.1 Initialization and longitudinal measurements

The detailed initialization experiment uses Pythia 2.8B with 32 transformer layers and 10,240 feed-forward neurons per layer. Indexing each neuron by its layer and activation coordinate gives 327,680 coordinates. The weights precede any optimizer update. The stimulus bank contains 1,216 item records: 1,088 scoreable records distributed evenly across 17 conditions, and 128 records whose task construction requires information unavailable at the prediction position. These latter records are excluded rather than assigned zero scores. Each measurable condition contains 64 items. The detailed comparisons use one initialization and one template block.

The longitudinal experiment uses Pythia 410M with 24 layers and 4,096 feed-forward neurons per layer, giving 98,304 coordinates. Discovery measurements cover every archived checkpoint from 1 through 165 on 54 tasks. An independently generated item set is evaluated at 24 selected checkpoints. It contains 32 items per task, or 1,728 unique items and 41,472 item–checkpoint measurements. The same items recur across checkpoints, allowing paired temporal comparisons. This archived trajectory does not include its own untrained checkpoint and is analyzed separately from the direct initialization experiment.

A.2 The first effective weight update

An optimizer call need not change the parameters. In the 2.8B experiment, the first call has a zero learning rate, and its saved weights equal the initialization. The following call produces the first nonzero displacement. The first-effective-update analysis uses these actual weight tensors. In the longitudinal experiment, checkpoint numbers identify archived states; the analysis compares the measured states rather than assuming that their numbering alone specifies every intervening optimizer operation.

B Measurement and statistical analysis

B.1 Prediction scores and neuron contributions

Let $s_y(x) = \sum_r \log p_\theta(y_r \mid x, y_{<r})$ be an answer's summed token log probability. The initialization experiment uses the correct-minus-foil contrast $s_{y^+}(x) - s_{y^-}(x)$. All answers in this stimulus set are single tokens, so summed and mean answer-token log probability coincide. Clean and corrupted prompts use the same answer identities. Attribution combines the clean–corrupted activation difference at the final prompt position with the derivative of this contrast evaluated on the corrupted prompt, as in Equation 2.

The 410M trajectory and its activation–sensitivity decomposition instead use $s_{y^+}(x)$ alone, with neither foil subtraction nor answer-length normalization. Attribution is evaluated at the final prompt position. Supplementary behavioral interventions use the correct-minus-foil contrast in mean answer-token log probability. These objectives answer different questions, so their numerical magnitudes are not pooled across experiments.

The complementary activation-gradient profile differentiates correct-answer negative log likelihood on the corrupted input and sums over causal-prefix positions. It is a gradient with respect to neuron activations, not model parameters. Its objective, sign convention, and position aggregation differ from the contrast derivative inside the attribution score.

B.2 Neuron selection and continuous comparisons

Signed item vectors are averaged within each condition in double precision before selection. Positive scores are ranked across all layers. The top 1% contains $K = 3,277$ neurons in the initialization experiment; all 34 groups, covering 17 conditions and two measurements, reach this size without a tied cutoff. Selected-neuron overlap is $|S \cap T|/K$, not the Jaccard index. Each altered condition is compared with its own baseline.

For full contribution vectors, cosine similarity summarizes direction independently of norm. This complements ranked overlap because scores close to the cutoff can exchange ranks while the full vector changes little. The measurements consequently distinguish continuous response changes from changes in selected membership.

B.3 Task-space directions and uncertainty

Let M_t contain one mean attribution vector per task at checkpoint t. Its Gram matrix $G_t = M_t M_t^\top$ captures task magnitudes and pairwise inner products. We examine both unscaled vectors and vectors divided by task-specific scales fixed from the discovery measurements. A task-space direction q gives the neuron loading vector $q^\top M_t$. Task rescaling requires a corresponding transformation of q to preserve the same contrast.

Squared sample means contain sampling noise. For event-window comparisons, we form mean vectors from two fixed, disjoint item-cluster subsets and estimate directional strength by

$$\widehat{E}_t = \frac{1}{N} \left\langle q^\top M_t^{(1)}, q^\top M_t^{(2)} \right\rangle, \qquad N = 98,304. \tag{3}$$

This cross-subset estimate can be negative and is not truncated. Changes compare the mean of three checkpoints before a selected boundary with the mean of three after it. We resample problem clusters within each task and fixed subset, using the same weights at all six checkpoints. Intervals are percentile intervals from 2,000 paired bootstrap replicates: approximate per-direction 95% intervals without family-wise correction. Random-subspace reference bands describe variation among directions rather than item-sampling uncertainty.

C Fixed-weight task manipulations

C.1 Separating input context from prediction targets

The template experiment replaces one, two, or three words in the surrounding frame while preserving each matched item's candidate answers and determiner corruption. The answer-vocabulary experiment makes the complementary change. Clean and corrupted prompts remain fixed, but the noun answers vary. Each condition draws from 12 noun lemmas, with 12, eight, four, or no lemmas shared with the baseline pool.

For example, the prefix "Morning has arrived. The teacher sees these" can be evaluated with the candidates "accounts" and "account", or with "baths" and "bath". The prefix is identical; the output contrast changes. At its final position, the forward activations therefore stay fixed while the output sensitivity can change. This comparison directly separates target dependence from a change to the input representation.

Table 2 reports all available baseline comparisons. Template changes and reduced answer-pool sharing produce graded responses under both measurements, although their magnitudes differ.

C.2 Sequence position and grammatical construction

The intervening-material manipulation relocates adjunct material within a subject–verb construction. This changes dependency distance together with word order, positions, and the final prompt token. The target-position experiment cyclically rotates a target clause and two neutral clauses; because scoring uses only the answer's causal prefix, its preceding context changes with position. The construction comparison changes a determiner–noun prediction to a subject–verb prediction, jointly changing the grammatical construction and answer category.

These sequence changes can affect the two measurements differently. For the intervening-material comparison, the attribution group contains 1,211 layer-0 neurons at baseline and none in the three modified conditions. The activation-gradient groups retain 1,778–1,856 layer-0 neurons. Thus the small attribution overlap does not imply that early-layer output sensitivity disappears.

The unchanged template baseline reuses the determiner–noun stimuli from the construction comparison. Their agreement checks consistency on shared stimuli rather than providing an independent repetition.

Table 2: Selected-neuron overlap (%) at unchanged initial weights. Every row compares a modified condition with its own baseline; both groups contain 3,277 neurons. Attribution combines activation differences and output sensitivity, while the gradient profile measures activation sensitivity to correct-answer loss.

Manipulation	Comparison with baseline	Attribution	Gradient
Template	1 replacement	91.33	91.30
	2 replacements	84.16	86.73
	3 replacements	77.02	81.72
Answer vocabulary	$2/3$ shared	54.01	51.36
	$1/3$ shared	32.65	32.16
	No shared answer lemmas	14.34	17.33
Intervening material	Level 1	6.07	65.43
	Level 3	6.07	65.49
	Level 5	6.10	65.40
Target position	Middle	66.46	71.86
	Late	58.29	65.79
Construction and target	Subject–verb vs. determiner–noun	6.13	12.39

D Cross-scale and cross-architecture comparison

Figure 3 combines the four Pythia scales and three other architectures on one categorical axis. Every model is evaluated with three template-word replacements and a disjoint answer-lemma pool. The selected fraction is fixed at 1% within each model. Table 3 gives the plotted values and the difference between template and answer overlap, rounded to one decimal place in percentage units.

Across the Pythia configurations, template overlap spans 73.0–80.0%, answer overlap spans 13.0–16.0%, and their difference spans 57.0–67.0 percentage points. Across the other architectures, the corresponding ranges are 71.0–77.0%, 16.0–20.0%, and 51.0–61.0 percentage points. The comparison highlights the separation between input-context and target changes, rather than ranking models by a single overlap value.

Table 3: Cross-model selected-neuron overlap (%) and the template-minus-answer difference in percentage points. These are the values used for the connected curves in Figure 3.

Model	Template overlap (%)	Answer overlap (%)	Difference (pp)
Pythia scale comparison			
Pythia 410M	73.0	16.0	57.0
Pythia 1.8B	76.0	15.0	61.0
Pythia 2.8B	78.0	14.0	64.0
Pythia 6.9B	80.0	13.0	67.0
Cross-architecture comparison			
OLMo-2 1B	75.0	18.0	57.0
BLOOM-3B	71.0	20.0	51.0
OpenELM-3B	77.0	16.0	61.0

The first four model categories compare parameter scales within Pythia; the remaining three compare other architectures. Parameter counts are included in the labels. Lines connect categories within each group to make the response profiles easier to compare, rather than defining a continuous scale across architectures. Each overlap is computed within its own model. The comparison therefore concerns the relationship between a task manipulation and neuron sharing, not a correspondence between individual neurons across networks.

E Training dynamics and supplementary analyses

E.1 Response along the first effective update

Let θ_0 denote the initial weights and θ_1 the weights after the first effective update. We evaluate

$$\theta(\alpha) = \theta_0 + \alpha(\theta_1 - \theta_0), \qquad \alpha \in \{-0.5, 0, 0.5, 1\}, \tag{4}$$

using the same stimulus bank and single-precision model evaluation. No optimizer update occurs during these measurements. For a condition mean μ, define $\Delta\mu = \mu(1) - \mu(0)$. The projection coefficient is $\langle \mu(\alpha) - \mu(0), \Delta\mu \rangle / \|\Delta\mu\|^2$. The residual measures deviation from $\alpha\Delta\mu$, relative to $\|\Delta\mu\|$. Table 4 reports medians across the 17 conditions.

Table 4: Local continuous responses along the first effective update. The positive midpoint closely follows the endpoint secant; negative extrapolation has larger deviations. The secant is defined by the measured endpoints rather than an independently estimated Jacobian.

Measurement	α	Projection	Direction cosine	Residual (%)
Attribution	0.5	0.499523	0.999152	2.06
Activation gradient	0.5	0.498597	0.998915	2.33
Attribution	−0.5	−0.495767	−0.992350	6.19
Activation gradient	−0.5	−0.492459	−0.990153	7.00

Pairwise cosine similarity follows the endpoint direction across the four evaluated states for 116 of 136 condition pairs under attribution and 123 under activation gradients. The corresponding top-1% overlap counts are 47 and 42. Among 12 prespecified focal pairs, mean cosine increases by approximately 3.0×10^{-4} under attribution and decreases by 7.2×10^{-5} under activation gradients. The update therefore changes different summaries differently; it does not uniformly increase task similarity.

E.2 Task-specific change over longer training

Event windows are selected using the discovery items and evaluated using independent items with the selected directions held fixed. Over checkpoints 66–71, one direction declines from a mean cross-subset strength of 1.33×10^{-4} to 6.37×10^{-5}. Its change is -6.96×10^{-5}, with a 95% interval of $[-1.07 \times 10^{-4}, -3.76 \times 10^{-5}]$. The companion direction has a positive point estimate but an interval spanning zero. The largest adjacent decline occurs from 67 to 68.

The other selected windows provide context. Over checkpoints 4–9, the change interval spans zero. The late window, 158–163, contains 12 fixed columns with a joint numerical span of approximately two dimensions; neither joint nor individual changes are resolved by the item samples.

For 128 selected neurons between checkpoints 68 and 69, the identity

$$\Delta(dg) = g_{\mathrm{pre}}\Delta d + d_{\mathrm{pre}}\Delta g + \Delta d\,\Delta g \tag{5}$$

separates activation-difference and output-sensitivity changes. The negative signed loading is dominated by $d_{\mathrm{pre}}\Delta g$. An alternative decomposition into mean products and within-task covariance attributes much of the same change to covariance. These are alternative descriptions of the selected-neuron signed change, not additive causal explanations of full-model strength.

E.3 Subspace coverage, rank, and behavioral checks

High coverage by an early task-space basis can reflect coordinate scale and dimension. A basis spanning 52 of 54 task coordinates already covers nearly the entire task space. As a post-hoc amplitude control, four task coordinates selected for early unscaled strength cover a median 85.16% of later independent-item strength, compared with 85.14% for the early four-dimensional principal subspace. High coverage alone therefore does not distinguish a shared low-dimensional mechanism.

A noise analysis uses two nonoverlapping samples of 6,144 neurons, with 256 sampled per layer. Across the 24 measured checkpoints, unscaled measurements usually support one detectable

dimension in both samples, whereas task-rescaled measurements are more sensitive to the sampled coordinates. These detection results concern the samples rather than the rank of the entire model.

Exploratory activation interventions around checkpoint 69 have effects unresolved by the item-level intervals, and their comparison groups differ substantially in baseline attribution strength. Small later-window effects do not restore positive correct-versus-foil margins. They therefore do not establish functional transfer between the groups.

E.4 Task alignment across item sets

Cross-set calculations align tasks by stable identifiers. Different row positions for 17 of the 54 tasks were corrected before combining new-item measurements with fixed scales or directions. Original item vectors were unchanged; affected derived comparisons were recomputed. Full-model cross-subset kernels agree with direct projections of the task means to within 7.3×10^{-12} across the three event windows.