Patches of Nonlinearity: Instruction Vectors in Large Language Models

Irina Bigoulaeva, Jonas Rohweder, Subhabrata Dutta, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science, Technical University of Darmstadt
and National Research Center for Applied Cybersecurity ATHENE, Germany
www.ukp.tu-darmstadt.de

Abstract

Despite the recent success of instruction-tuned language models and their ubiquitous usage, very little is known of how models process instructions internally. In this work, we address this gap from a mechanistic point of view by investigating how instruction-specific representations are constructed and utilized in different stages of post-training: Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO). Via causal mediation, we identify that instruction representation is fairly localized in models. These representations, which we call *Instruction Vectors* (IVs), demonstrate a curious juxtaposition of linear separability along with non-linear causal interaction, broadly questioning the scope of the linear representation hypothesis commonplace in mechanistic interpretability. To disentangle the non-linear causal interaction, we propose a novel method to localize information processing in language models that is free from the implicit linear assumptions of patching-based techniques. We find that, conditioned on the task representations formed in the early layers, different information pathways are selected in the later layers to solve that task, i.e., IVs act as *circuit selectors*.[1]

1 Introduction

Instruction tuning has emerged as a staple of post-training in recent years, with some form of it now present in nearly all state-of-the-art LLMs. Originally conceived as a supervised fine-tuning (SFT) technique (Ouyang et al., 2022; Wei et al., 2022), instruction tuning has since been implemented in many variations, which range from alternate loss calculations (Shi et al., 2024; Chatterjee et al., 2025), to using adapters (Liao et al., 2024) or preference optimization (Bi et al., 2024). Prior works have investigated the differences between base and instruction-tuned models in terms of output patterns and task performance (Zhou et al., 2023; Ghosh et al., 2024). Fine-tuning in general (which is the most commonplace implementation of instruction tuning) has been shown not to introduce new capabilities into a pretrained model, rather reusing existing ones (Jain et al., 2024).

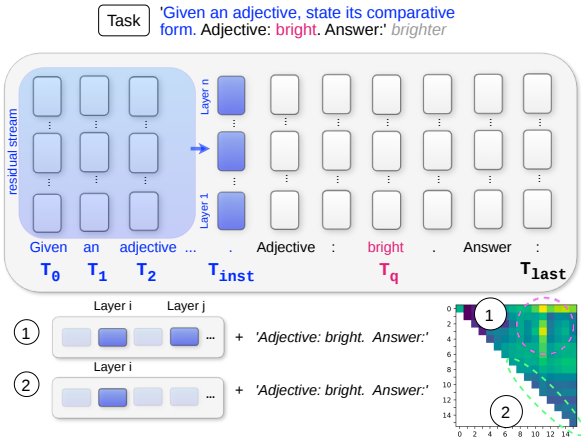

Figure 1: We locate instruction vectors (IVs) in the residual stream representations of the final instructional token, T_{inst}. Across models and tasks, we find that T_{inst} stores sufficient instruction information, and that layer-wise representations are more effective in combination (1) than alone (2), i.e. IVs are *superadditive* (§ 5).

However, one fundamental question remains unaddressed: By what mechanism do models represent and process instructions? Isolating this mechanism will enable an accurate diagnosis of whether a model has successfully acquired the ability to follow instructions for a given task, and which post-training methods (if any) are necessary for this capability to appear. Furthermore, instruction-following is a pillar of alignment; safer, accountable deployment of LLMs necessitates a thorough understanding of how instructions are processed by the language model.

Prior studies identify that models construct localized neural digests of input-output mappings from

[1]We make our code available at: https://github.com/UKPLab/acl2026-instruction-vectors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12209–12262
July 2-7, 2026

in-context examples (Hendel et al., 2023; Todd et al., 2024; Li et al., 2025). These digests, commonly denoted as *function vectors*, restore the task context when provided during a new forward pass, even without any in-context examples. Recently, Davidson et al. (2025) demonstrate that similar transferrable digests can be associated with instructions as well, realized at different yet overlapping attention heads of the model.

We formulate our starting point motivated by these works, and note the following gaps: 1) Function vectors, identified at the last token of the input prompt, are the end-products of the internal computation, but the process of their formation, geometric properties, and mechanism of action remain unknown; 2) Steering vector approaches (either in the model activation space (Stolfo et al., 2025) or in SAE feature space (He et al., 2025)) identify optimal perturbation locations that elicit superior instruction following, albeit without providing additional knowledge about it.

Contributions and findings. In this work, we seek to provide a mechanistic exploration of instruction following by investigating how models represent instructions. We discover that models construct localized digests, representative of the instruction, right after processing the instructional context (Section 5). The task information in these digests is linearly recoverable (i.e., there exist linear hyperplanes that discriminate the digests corresponding to semantic paraphrases of two tasks). However, the action of these digests is *superadditive* in nature: there exists a combination of layers whose cumulative causal influence is more than the sum of their individual influences. This superadditivity signals a non-linear interaction between the linearly-separable digests. This limits the applicability of existing causal discovery methods, which presume an additive interaction between the variables, for investigation of how these instruction digests are utilized.

To mitigate this challenge, we propose a novel method of understanding information flow within a Transformer-based language model (Section 6.1) that is free from additive interactions between model components. Using this method, we identify that the neural digests play the role of *circuit selectors* to process the query. While a pretrained model can construct the same task-specific neural digests, instruction-tuning is indispensable for introducing the circuit-selection ability into those digests.

Broader significance. The fact that localized instruction digests are constructed independent of the query (i.e., eager computation instead of just-in-time computation together with the query) points towards the possibility of information bottleneck (Tishby et al., 2000) in instruction processing. Such a bottleneck can be utilized to increase robustness against adversarial perturbations in the instruction. Superadditivity of task digests bears foundational implications for interpretability research: If the influence of a variable X depends on another, non-local variable Y, then single-component causal attributions can undervalue X by recording a low impact of X when Y is not active. This calls for an immediate evaluation of how we think of mechanistic interpretability models.

2 Related Work

Several recent works have investigated how models represent tasks. Todd et al. (2024) find that models form compact representations of tasks, specifically from in-context learning (ICL) examples. Davidson et al. (2025) extend this investigation to instructions and find that ICL- and instruction-based task representations activate different attention heads, but are beneficial when used together. An orthogonal yet noteworthy study by Huang et al. (2024) represents instruction-following as differential directions in the parameter space. Additionally, various works have examined the properties of instruction-tuned models, from the standpoint of behavioral differences to base models (Wu et al., 2024) and whether a particular dimension can be isolated that yields instruction-following behavior (Heo et al., 2024). Nevertheless, a gap is left in this research area, namely, how models form instruction representations and what the causal and geometric properties of these representations are.

A substantial majority of mechanistic interpretability research implicitly or explicitly associates abstract causal roles with singleton components (representation, subspace, parameter matrix such as attention heads, etc.): name-mover heads (Wang et al., 2023), function vectors (Todd et al., 2024), factual association in early-middle MLPs (Meng et al., 2022), etc. Circuit discovery (Wang et al., 2023; Conmy et al., 2023; Ameisen et al., 2025) in its current form is built on top of this assumption – these algorithms formalize the forward pass of a model as a Directed Acyclic Graph (DAG) and search for a connected subgraph responsible for the behavior under inves-

tigation. Findings of Sutter et al. (2025) warn of the fallibility of causal abstraction without strong assumptions on how language models encode features.

A current popular theory of model representations is the *linear representation hypothesis*, which states that models represent "high-level concepts" as linear subspaces (Elhage et al., 2021; Sharkey et al., 2025; Gurnee and Tegmark, 2024; Nguyen and Leng, 2025). Theoretical frameworks have additionally been developed that seek to formalize the ways in which concepts interact linearly within a model's representation space (Park et al., 2024; Nguyen and Leng, 2025). Our findings identify a critical orthogonality in this area: despite linear representation, causal variables can interact synergistically, rendering the DAG model of causality invalid (since a DAG allows a causal edge to connect two nodes, it cannot capture the scenario when two variables synergistically define a third variable). Subsequently, our proposed solution, via locally linear maps, searches for a suitable subset of the functions jointly implemented by a Transformer, without any component-specific abstraction. Note that our results go beyond the existing argument related to sparse vs. distributed causal structures (Bahador, 2025). Even in a distributed setup, a single abstraction of the causal variable is shared across multiple layers/locations (Lindsey et al., 2024). Each instance of the distributed variable interacts additively, whereas our discovery points toward a completely different organization of causal variables that promote superadditivity.

3 Models and Tasks

Models. We use OLMo-2 (OLMo et al., 2025) models due to their clear progression from the base, SFT, and DPO variants – e.g. the SFT model was built from the base model, and the DPO model was built from the SFT model. This enables us to follow the impact of the typical post-training process when comparing the models.

We implement our experiments using NNSight (Fiotto-Kaufman et al., 2025), setting library hyperparameters to ensure deterministic behavior and reproducibility. We elaborate on GPU architecture and experiment runtimes in Appendix A.

Tasks. We generate two *contrastive task pairs* – i.e. pairs of tasks in which the target query is the same, while the instructions differ (see Table 1). In each case, the model processes an identical query but must output a different token as the answer. This is done to minimize the variation in the prompts apart from the instruction and follows the task design of similar works (Hendel et al., 2023). We generate these tasks using ChatGPT-5-nano (OpenAI, 2025) and manually evaluate the correctness of each sample (∼200 samples per task).

Additionally, to show generalization, we use four tasks from the BigBench benchmark (Srivastava et al., 2023): METAPHOR_BOOLEAN, IMPLICATURES, OBJECT_COUNTING, and SNARKS.

Confirming Instruction Following Ability. When investigating instruction representations, an important confounding factor to consider is the model's competence in a given task. If a model is unable to solve a task, then it is reasonable to suppose that it will also not have instructional representations for that task, or that the properties of these representations will differ from those of a more capable model. Thus, we evaluate each model's performance on these tasks using exact match accuracy (EMA).

Additionally, we establish a metric called INSTRUCTIONAL ACCURACY (IA), which measures whether the model's answer falls within a manually-defined scope of valid response types for the instruction, even if its output answer is incorrect. A detailed elaboration of this metric can be found in Appendix C.

We find that most models are able to follow instructions for most tasks with a high degree of accuracy (IA $>$ 50%, see Figure 20 in Appendix C). This confirms that our choice of tasks is appropriate for the models, and implies that the models can in fact represent and process instructions for these tasks.

4 Preliminaries

In this section, we provide the theoretical preliminaries for our experiments.

Let $\mathbb{V}$ be an indexed vocabulary of tokens where every token $v \in \mathbb{V}$ is represented as the standard column basis vector $\boldsymbol{e}^{(v)} \in \mathbb{R}^{|\mathbb{V}|}$ (i.e., the v-th element is 1, the rest are 0). An n-length input token sequence is defined as $\boldsymbol{T} := T_1, \cdots, T_n \in \mathbb{V}^*$. The forward pass computation of the l-th layer of an auto-regressive transformer with H attention heads, for $l \in \{1, \ldots, L\}$, can be written as fol-

Task	Subtask	Instruction
Adjectives	Comparative	'Given an adjective, state its comparative form.'
	Antonym	'Given an adjective, state its antonym.'
Animals	Color	'Given an animal, state its most typical associated color.'
	Can_Fly	'Given an animal, state whether or not it can fly. Print 'yes' or 'no'.'
BigBench	Metaphor_Boolean	'For a given metaphorical sentence, identify if the second sentence is the correct interpretation. Print Y for yes and N for no.'
	Implicatures	'Predict whether Speaker 2's answer to Speaker 1 counts as a yes or as a no. Print Y for yes and N for no.'
	Object_Counting	'For the given sentence, count the number of objects listed and print it as a digit.'
	Snarks	'Choose which of the two statements is sarcastic and print the corresponding letter:'

Table 1: Our eight tasks with their corresponding instructions. During tokenization, the final token is always the final punctuation mark (either a fullstop or a colon).

lows:

$$
\begin{aligned}
X_i^{l,\text{att}} &= \sum_j a_{i,j}^{l,h} \boldsymbol{W}_{OV}^{l,h} X_j^1 \\
\boldsymbol{X}^{l,\text{mid}} &= \text{Norm}(\boldsymbol{X}^{l,\text{att}} + \boldsymbol{X}^l) \\
\boldsymbol{X}^{l,\text{mlp}} &= \boldsymbol{W}_2 \,\text{PLin}\left(\boldsymbol{W}_1 \boldsymbol{X}^{l,\text{mid}}\right) \\
\boldsymbol{X}^{l+1} &= \text{Norm}(\boldsymbol{X}^{l,\text{mid}} + \boldsymbol{X}^{l,\text{mlp}})
\end{aligned} \tag{1}
$$

where $\boldsymbol{X}^1, \boldsymbol{X}^{l,\text{att}}, \boldsymbol{X}^{l,\text{mid}}, \boldsymbol{X}^{l,\text{mlp}} \in \mathbb{R}^{n \times d}$ are the most commonly referred-to model representations, $h \in \{1, \ldots, H\}$ is the index of the attention head, and $a_{i,j}^{l,h} \in \boldsymbol{a}_i^{l,h}$ is attention between i-th query and j-th key. We follow Elhage et al. (2021)'s reparameterization of multi-head attention: For each attention head, the value and output projections are folded into a single transformation $\boldsymbol{W}_{OV}^{l,h} \in \mathbb{R}^{d \times d}$. Norm denotes learnable normalization operation (LayerNorm or RMSNorm). Parameters $\boldsymbol{W}_1^l \in \mathbb{R}^{d' \times d}$, $\boldsymbol{W}_2^l \in \mathbb{R}^{d \times d'}$, and piecewise-linear function PLin() (e.g. ReLU, GELU) together construct the MLP block.

The input to the first decoder block is computed as $\boldsymbol{X}^1 = \boldsymbol{X}^{\text{emb}} = \boldsymbol{W}_E \boldsymbol{T}$, where $\boldsymbol{W}_E \in \mathbb{R}^{d \times |\mathbb{V}|}$ is canonically called the embedding that maps each token to the d-dimensional representation space. The final representation, i.e., $\boldsymbol{X}^{L+1}$ is mapped to the token logit space via multiplying by $\boldsymbol{W}_U$ (commonly referred to as the unembedding projection).

Unless mentioned otherwise, the input token sequences in our setup can be formalized as $\boldsymbol{T}^{\text{full}} := [\boldsymbol{T}^{\text{inst}} \boldsymbol{T}^{\text{q}}]$, where $\boldsymbol{T}^{\text{inst}}$ is the instruction describing the task and $\boldsymbol{T}^{\text{q}}$ is the query information. Since we are primarily concerned with the token at the very end of the instruction segment and the last token of the prompt (See Section 5), we refer to these token positions as T_{inst} and T_{last}, respectively. See Figure 1 for an illustration of our approach.

Finally, we denote the abstract forward pass of the model as

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{T}; \Theta)$$

where Θ denotes the set of interventions (on representation or parameter space of the model) applied. For example, switching off the h-th attention head on l-th layer can be represented as $\Theta := \boldsymbol{W}_{OV}^{l,h} \leftarrow \boldsymbol{0}$.

5 Localizing Instruction Representations

We start with posing a fundamental question about the mechanism of instruction following: Are instructions processed *just-in-time* (i.e., simultaneously or after processing the query) or *eagerly* (i.e., model constructs neural digests of the instruction before processing the query)? The former would be realized with distributed computation within the model, whereas the latter would result in localized representations of the instruction.

We start with causal variable discovery using activation patching (Zhang and Nanda, 2024) to check for localizability of the instruction representations before the query tokens. Note that our primary focus in this stage is to determine the synthesis location of instruction digests (if any exist) and not their downstream usage mechanism. Therefore, we restrict ourselves to the residual stream representations ($\boldsymbol{X}^l$ in our formalism) instead of finer granularity like attention head or MLP neuron outputs.

5.1 Causal Mediation Analysis

Our adopted causal variable localization strategy closely follows prior work by Meng et al. (2022). For any given internal variable $\boldsymbol{X}^l$, causal media-

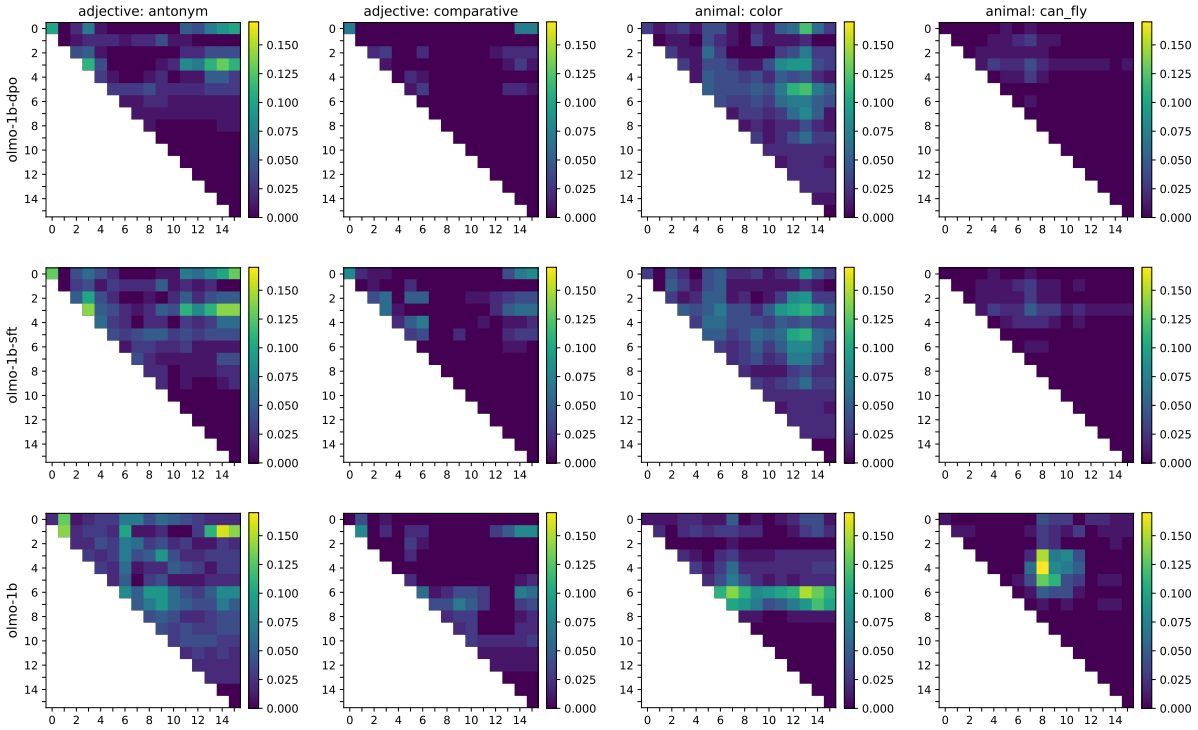

Figure 2: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching. Two important properties are shown. *Localization:* Across all tasks, we observe localized points where the logit improvement is the greatest. *Superadditivity:* Two-layer combinations bring greater improvements than single layers. Generalization to 7B models and other tasks is shown in the appendix.

tion starts with two distinct forward pass:

$$\boldsymbol{y}^{\text{source}} = \mathcal{F}\left(\boldsymbol{T}^{\text{full}}; \emptyset\right) \quad \boldsymbol{y}^{\text{target}} = \mathcal{F}\left(<s> \boldsymbol{T}^{\text{q}}; \emptyset\right)$$

where $< s >$ is a filler token. Next, we run the forward pass under intervention:

$$\boldsymbol{y}^{\text{patch}} = \mathcal{F}\left(\boldsymbol{T}^{\text{q}}; \boldsymbol{X}^{l}_{<s>} \leftarrow \boldsymbol{X}^{l,\text{source}}_{T^{\text{inst}}}\right)$$

The causal effect of the variable $\boldsymbol{X}^l$ can then be quantified as $d\left(\boldsymbol{y}^{\text{patch}}, \boldsymbol{y}^{\text{target}}\right)$, where d is suitable metric of choice, conditioned on the answer token. Specifically, we use two metrics: (1) difference between the reciprocal rank of the answer token, and (2) difference between the absolute logit values of the answer token[2], in $\boldsymbol{y}^{\text{patch}}, \boldsymbol{y}^{\text{target}}$.

In summary, our setup checks for localizability of instruction digests by testing whether the model can recover the correct answer from the query context and a small subset of the instructional representations.

Superadditivity of IVs. We conduct single- and multi-layer activation patching[3]. For OLMo-2 1B models, Figure 2 shows the effects of 1- and 2-layer patching on the reciprocal rank of the target token, while Figure 10 shows the effect on the logit of the target token. The scale is normalized to better elucidate differences between the base and post-trained checkpoints; unnormalized heatmaps are presented in Appendix B.

We find that the greatest improvements in reciprocal rank tend to occur in localized areas – for example, in the layer combinations (3, 7) and (3, 4) for the ANIMAL: CAN_FLY task (see Figure 2). However, when these individual layers are patched by themselves, the relative rank improvement is much lower. The same is observed with the OLMo-2-7B models for both rank and logit (see Figures 6 and 7 of the appendix).

In other words, we find that the instructional representations at various individual layers have a greater effect on the rank and logit of the tar-

[2]Our localization is primarily based on reciprocal rank. Logit difference alone can be confounded due to high-norm representations that simply increase the logit value of all tokens.

[3]We describe multi-layer settings as tuples, e.g. (1, 2), but since the layers are patched simultaneously, the tuple ordering does not matter. In other words, (1, 2) is the same as (2, 1). Figure 2 and the other heatmaps are accordingly left blank in the redundant grid areas.

get token when acting together in combination, rather than when their individual contributions are summed together. In abstract terms:

$$f(\boldsymbol{x}_i \cup \boldsymbol{x}_j) \geq f(\boldsymbol{x}_i) + f(\boldsymbol{x}_j) \quad (2)$$

for two given causal variables $\boldsymbol{x}_i, \boldsymbol{x}_j$, and f conceptually denotes the causal effect function implemented by the model.

To more rigorously test whether this inequality holds, we perform a t-test on the numerical scores. We find that, for all models and tasks, the inequality indeed holds to a significant degree across samples (see details and results in Tables 5-8 in Appendix B). This result indicates that the mechanism of L_i and L_j's interaction is not a linear operation (e.g. addition), but rather one that is nonlinear. In other words, instruction vectors are *superadditive*.

Base vs. Post-Trained Checkpoints. Since we observe superadditivity across all model checkpoints, it is likely that the multi-component nature of instructional representations is a fundamental property, and that the role of post-training is not to change their nature, but rather to refine them. In particular, we notice that the post-trained models tend to form representations in similar layer groups, but at different ones than the base model (see Figure 2 and Appendix B), and that they are more "spread-out" than those of the base model. This implies that post-trained models rely on more complex multi-component interactions. Model size is also an influencing factor: In the 7B models, the post-trained checkpoints have higher logit and rank improvements than the base model, while the 1B models have a noisier variation among the checkpoints.

Geometric Properties of IVs. Finally, we wish to determine how the instructional representations of our eight tasks are geometrically configured in the models' representation space. For each of our contrastive tasks, we produce 200 rephrasings of the instruction to test robustness against within-task variation. We generate these using ChatGPT-5-nano (OpenAI, 2025) and manually check each sample for correctness. For each instruction, we extract the residual stream representations using the same procedure as in Section 5.

We plot the LDA representations of our models and find that the instructional representations form well-defined task clusters, albeit ones that are not linearly separable in 2 dimensions (see Appendix D). These results partially contrast with

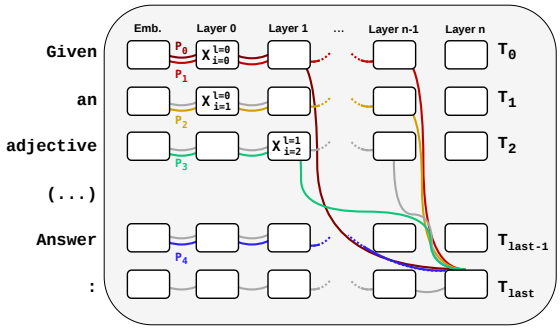

Figure 3: Conceptual decomposition of the Transformer as a collection of locally-linear, token-to-token maps, indicating how information flows through the model. For each layer and token position, high-ranking paths (in color) to the output token may exist. Other paths (in gray) may be low-ranking, or may not lead to the target token.

prior works, which found task clusters that were linearly separable in two-dimensional TSNE representations (Hendel et al., 2023).

To confirm whether or not our task clusters are linearly separable in general, we train a simple linear probe on a portion of the representations and attempt to predict the task cluster given a test sample. We find that our linear probe has high accuracy for all task clusters (98%<). This implies that the instruction vectors do form linearly separable clusters, but potentially at a high dimension that was not captured by our dimensionality reduction method. Furthermore, the fact that this separation is present in both base and post-trained checkpoints implies that post-training does not newly introduce this geometric property: Base models also distinguish between instructions of different tasks.

6 Causal Mechanism of Instruction Vectors

In the previous section, we demonstrated the localization of instruction vectors and showed that these representations are indeed used for instruction following. However, it remains unclear *how* exactly they are utilized.

Existing methods of discovering causal mechanisms (i.e. circuit discovery) rely on repeated, component-wise causal mediation to disentangle the task-specific causal graph within the model (Wang et al., 2023). However, the distributed, synergistic interaction of instruction vectors imply that such component-induced graphical models will produce misleading mechanisms.

Locally-linear surrogate maps. To circumvent

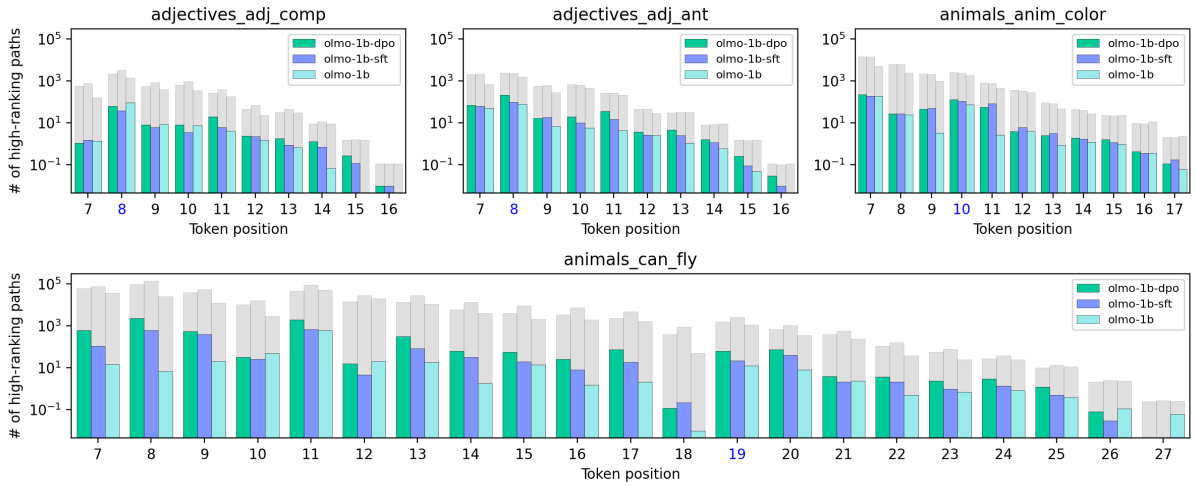

Figure 4: Path contribution by token position for 1B models. For each task, we examine a subset of token positions in the prompt, and for each token position, we average the number of high-ranking paths over 100 task samples (in color). The average total number of paths over the 100 samples is shown in gray. The T_{inst} tokens for each prompt are highlighted in blue. Plots highlighting the percentage contributed by each token are shown in Appendix E.

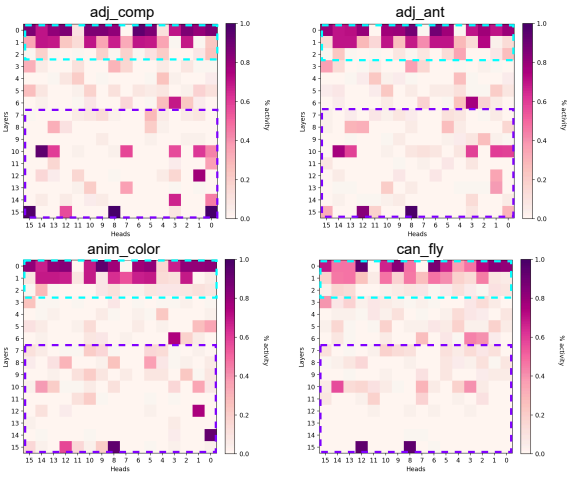

Figure 5: Attention head activity for OLMo2-1B across the contrastive tasks. The values represent how often a certain head is active (% over 100 samples, significant). We highlight the areas where the head activity is similar (blue) and diverges (purple). Note the similarity at the T_{inst} tokens for ADJECTIVES tasks, and an earlier token of ANIM: CAN_FLY, which has a multi-sentence instruction (see Section 6.1).

this, we propose a novel, intervention-free method to attribute the components with task-specific information propagation. In this approach, we represent the Transformer as a collection of interacting, token-token, locally-linear maps. Note that in the absence of the MLP layers, Elhage et al. (2021) already provide reparameterization of 1- and 2-layer attention only transformers as linear maps in $\mathbb{R}^{|\mathbb{V}|\times|\mathbb{V}|}$. To handle the intermediate non-linear operations such as normalization and MLP, we draw motivation from the neural polytope literature (Black et al., 2022; Hanin and Rolnick, 2019). For any given layer l, we can rewrite the MLP operation as:

$$X_i^{l,\text{mlp}} = \boldsymbol{V}_i^l X_i^{l,\text{mid}} = \boldsymbol{W}_2 \boldsymbol{D} \boldsymbol{W}_1 X_i^{l,\text{mid}}$$

where $\boldsymbol{D} \in \mathbb{R}^{d' \times d'}$ is a diagonal matrix that is a surrogate of $\text{PLin}()$ at the point $\boldsymbol{W}_1 X_i^{l,\text{mid}}$. For example, if $\text{PLin}()$ is ReLU, then

$$diag(\boldsymbol{D}) := \mathbb{1}_{\boldsymbol{W}_1 X_i^{l,\text{mid}} > 0}$$

Similarly, normalization can be 'linearized' in this manner by treating it as an affine transformation determined by the input mean and standard deviation, resulting in a similar surrogate $\boldsymbol{U}_i^l$. Note that, while these surrogates allow treating the non-linearities as linear maps, they are restricted within an ϵ-neighborhood of the current input. Using these surrogates, we can rewrite the l-th layer operation (Eq. 1) as:

$$\begin{aligned} X_i^{l+1} &= \boldsymbol{U}_i^{l,\text{mlp}} \left(\boldsymbol{I} + \boldsymbol{V}_i^l\right) \boldsymbol{U}_i^{l,\text{att}} X_i^l \\ &+ \boldsymbol{U}_i^{l,\text{mlp}} \left(\boldsymbol{I} + \boldsymbol{V}_i^l\right) \boldsymbol{U}_i^{l,\text{att}} \sum_h \sum_j a_{i,j}^h \boldsymbol{W}_{h,OV} X_j^l \end{aligned} \tag{3}$$

Structurally, this corresponds to a sum of linear transformations, each corresponding to $2(H+1)$-many different paths from input to output. By stitching together similar paths across layers, we represent the forward pass as a sum of multiple token-to-token locally-linear maps (see Figure 3).

6.1 Determining the Mechanism

Instead of the component configuration space (i.e., set of all connected subgraphs of the computation graph), we use its dual, token-token function space to localize and characterize the causal mechanism. Since no interventions are applied and interacting paths (i.e., two paths sharing $\boldsymbol{V}_i^l$) are allowed to use the shared channel, the effects of these paths remain additive, although instruction vectors themselves are not.

However, such a function space grows combinatorially with size of the model and input token sequence, making it infeasible to examine each existing path. Thus, for computational practicality, we trace only those paths that gather maximal attention, i.e., we consider an edge (via attention head h) between X_j^l and X_i^{l+1} only when

$$j = \arg\max \boldsymbol{a}_i^{l,h}$$

For each path $\boldsymbol{P}$ that maps input token T_i to the output, we can compute the logit contribution as $\boldsymbol{W}_U \boldsymbol{P} \boldsymbol{E}_E T_i$. Subsequently, we consider paths that result in answer token rank < 100.

We use these token-token maps to conduct three angles of analysis, each of which reveals an aspect by which instructional representations are utilized. Due to memory constraints, we conduct these experiments on OLMo 1B models.

Path Contribution by Token. A general observation is the strong sparsity of the traced paths: despite a combinatorial possibility, the number of high-ranking paths (<100) per token remains bounded by $\mathcal{O}(10^3)$. This high degree of sparsity implies the practical appeal of our proposed path-tracing for causal discovery. Any path emanating from a token closer to the last input token has fewer choices of next node compared to distant ones. Therefore, random distribution of path geometry would result in a strictly decreasing total number of paths as we go from first to last input token. Yet, across tasks and model variations, we find that T_{inst}-emitted high-ranking paths form the upper bound among other tokens (Figure 4). This indicates that T_{inst} indeed plays a crucial role in the correct processing of an input instruction, leading to the target output token. However, the exception to this pattern is equally informative: In ANIM: CAN_FLY, the token position with the most high-ranking paths is T_{11}, which occurs before T_{inst}. This is explainable due to the greater complexity of this task instruction: *"Given an animal, state whether or not it can fly. Print 'yes' or 'no'."* T_{11} represents the fullstop at the end of the first sentence, while T_{inst} is the final fullstop. In fact, the first fullstop already forms a complete instruction, while the second sentence supplements it. This reinforces the eager nature of instruction representations and suggests that the models prefer to utilize a complete representation of the sub-instruction ending at T_{11}.

Attention Head Activity. Next, we investigate whether the information-processing paths elicited by different instructions differ structurally (i.e., different instructions = different circuits). We use *attention head activity* along the high-ranking paths emitted from T_{inst} as the proxy of instruction-specific circuit structures. Concretely, for a given task, we record the fraction of times a particular attention head participates in a high-ranking path. This measure indicates a structural summary of the causal mechanism post-instruction.

We enumerate the paths as follows: For each query token, we take the top-k most attended-to tokens, first with k=1, then k=2. Within each k setting, we calculate a particular head's mean rate of activity over a sample size of $n = 100$, along with bootstrapped confidence intervals with 10,000 resamplings. Additionally, we calculate the variance of each head's activity against the Gaussian distribution using a t-test, to determine whether the head's activity pattern is significantly different from noise. Figures 25 - 32 in Appendix F.1 provide a visualization.

Across all models, we find that most heads have significant non-noise activity. The remaining heads are mainly deterministic (always on or off); noise heads are very rare. With increasing k, more significant heads appear, and head activity increases overall. This is to be expected, since more paths are added to the computation graph.

Despite the increase in total head activity, we notice a pattern that stays robust across k: namely, that earlier layers have similar active heads, while later ones differ. We quantify this by computing the Jaccard similarity of the numerical matrices corresponding to the heatmaps, between both k settings (shown in Table 3).

Figure 5 provides a visualization of these patterns. For ADJ: COMPARATIVE, ADJ: ANTONYM, and ANIM: COLOR, the head activity pattern remains nearly identical in the earlier layers (particularly 0-1), then begins to differ in the later layers between the tasks. Interestingly, we observe that

Spearman r per layer (k=1 vs k=2)																
Adjectives: Antonym																
Model	**0**	**1**	2	3	**4**	**5**	6	7	**8**	9	10	**11**	12	13	**14**	**15**
olmo-1b	**0.65**	**0.63**	0.34	0.32	0.08	0.40	0.43	0.28	**0.65**	0.47	0.50	-0.18	-0.14	0.07	0.44	0.59
olmo-1b-sft	**0.80**	0.47	0.31	-0.10	**0.86**	**0.80**	0.52	0.49	0.24	-0.32	0.28	0.57	-0.17	0.07	**0.62**	**0.80**
olmo-1b-dpo	**0.79**	0.50	0.24	-0.04	**0.72**	**0.87**	0.25	0.45	0.36	-0.11	0.09	**0.64**	0.00	-0.02	0.44	**0.79**
Adjectives: Comparative																
Model	**0**	**1**	2	3	**4**	**5**	6	7	8	9	10	**11**	12	**13**	14	**15**
olmo-1b	0.58	**0.61**	0.41	0.15	0.23	**0.60**	0.39	0.15	0.51	0.29	0.59	0.27	0.26	**0.60**	0.37	**0.69**
olmo-1b-sft	**0.78**	0.37	0.36	-0.14	**0.79**	0.21	0.52	0.56	0.42	0.07	0.11	**0.61**	0.07	0.57	0.10	**0.74**
olmo-1b-dpo	**0.77**	0.42	0.28	0.13	**0.71**	0.42	0.49	0.59	0.51	0.26	0.27	**0.61**	0.22	0.10	0.02	**0.84**

Table 2: Spearman r correlations between k=1 vs k=2 settings for the ADJECTIVES tasks. Layers containing heads with a strong positive correlation (≥ 0.6) for at least one model are shown in color, and the corresponding values are bolded. The significantly active layers correspond roughly to the locations of IVs found in Section 5.

for ANIM: CAN_FLY, the attention head activity patterns at T_{11}^{fly} most closely resemble those for ANIM: COLOR at $T_{10}^{\text{color}} = T_{\text{inst}}^{\text{color}}$. Meanwhile, the head activity patterns for $T_{19}^{\text{fly}} = T_{\text{inst}}^{\text{fly}}$ are markedly different from both of these token positions. Correspondingly, T_{10}^{fly} was the token position with the most high-ranking paths for ANIM: CAN_FLY.

	adj_comp			adj_ant		
Model	L0–1	L4–5	L14–15	L0–1	L4–5	L14–15
OLMo-1B	**0.91**	0.25	0.24	**0.88**	0.18	0.38
OLMo-1B-SFT	**0.84**	0.26	0.48	**0.81**	0.19	0.43
OLMo-1B-DPO	**0.84**	0.26	0.35	**0.81**	0.27	0.36

Table 3: Jaccard similarity of active heads (threshold > 0.1) between k=1 and k=2, by layer group at T_{inst}. Generalization to k=3 is shown in Appendix F.2.

Active Heads and Instruction Vectors. The observed head activity patterns have important connections to instruction vectors. Namely, we notice that the layers which contain significantly active heads (Table 2) for a particular model and task are also the layers that are involved in most of the 2- or 3-layer tuples that constitute instructional representations, presented in Section 5.

For example, in Table 18, we see that most top layer triples for the post-trained models in the ADJECTIVE: ANTONYM task involve Layers 14 or 15, which are strongly active across k. Meanwhile, for the base model, most triples involve Layers 0 and 1. Additionally, while inactive layers such as 14 and 15 are sometimes involved in the top triples, they tend to appear alongside a strongly-active layer.

A similar trend holds for ADJECTIVE: COMPARATIVE. Here, Layer 11 appears most often in the top layer triples for SFT and DPO, while it appears only 3 times for the base model, and always in combination with the more active Layer 15. We do note minority cases where top-contributing tuples consist solely of weakly active heads – however, these tuples tend to be tied with the same score as tuples containing highly active heads. This implies that further multi-layer interaction might be occurring.

Overall, we observe that *instruction vectors corresponding to different tasks are constructed in the early layers* using nearly identical, instruction-agnostic circuits (similar head activity), *followed by task-specific circuits that are conditioned by the instruction vector* (different head activity).

7 Conclusion

In this work, we examine the internals of language models to gain insight into the mechanisms of instruction following, comparing the base model with its post-trained counterparts. We causally locate neural digests of task representation, which are formed in an eager manner right after the instruction is processed. While these digests are linearly separable according to task semantics, they interact with each other in a nonlinear, synergistic manner. This finding provides counter-evidence against the assumption of 'one component, one causal role', commonplace in the current landscape of mechanistic interpretability (Wang et al., 2023). Subsequently, this makes computational graphs unreliable for identifying the causal mechanisms of instruction processing. To mitigate this challenge, we propose a novel method of tracing information flow within Transformers that is free from additive assumptions, allowing us to pinpoint the importance of each model component in processing an instruction. Using this approach, we identify the role of instruction vectors as circuit selectors.

Limitations

Due to memory constraints, our experiments regarding the causal mechanism are conducted on 1B models. However, these models may be limited in their general task-solving capabilities compared to 7B models. While the general trend of our conclusions remains the same, more elaborate patterns may emerge at larger model scales. Additionally, while our results regarding the base, SFT, and DPO checkpoints provides initial insight into how instruction representations evolve during the course of model training, a finer-grained analysis on a greater number of intermediate checkpoints (e.g. with Pythia models (Biderman et al., 2023)) could be conducted.

Our definition of contrastive task pairs is based upon human-defined distinctions - i.e. we give differing instructions while keeping the target query the same. Nevertheless, this does not guarantee that an LLM will also perceive these tasks to be different. An analysis could be conducted that involves contrastive instructions as determined by an LLM's notion of task similarity, e.g. defined using distance metrics between individual sample representations. However, conducting such an analysis is beyond the scope of this work.

Similarly, we choose instructions which we consider to be fairly simple, both in terms of their linguistic structure as well as in the underlying task that they represent. Studying the impact of task difficulty and instructional complexity is worthwhile, but is beyond the scope of this work due to the significant theoretical and experimental foundation required (i.e. providing a satisfactory definition of "difficulty" and conducting a thorough ablation).

Finally, the instructions that we examine in this work all have a clear syntactic separation between the instruction and query. This is purposefully done to make it possible to investigate the point in the prompt when the instruction is fully processed, but the query hasn't yet been. This separation is also important for our path analysis experiments, since the case where the query can be cleanly switched out across samples without altering the instruction results in the fewest confounding factors. However, we acknowledge that real-world instructions are often intermixed, and may yield additional interesting insights.

We acknowledge these limitations with the intention that our work may build a starting foundation for future research in this area.

Acknowledgements

This work was funded by the LOEWE Distinguished Chair "Ubiquitous Knowledge Processing", LOEWE initiative, Hesse, Germany (Grant Number: LOEWE/4a//519/05/00.002(0002)/81), as well as by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

We would like to thank Prof. Kentaro Inui and Phu Hoang for the insightful and helpful discussions regarding an early version of this paper.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L Turner, et al. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Nooshin Bahador. 2025. Localized definitions and distributed reasoning: A proof-of-concept mechanistic interpretability study via activation patching. *CoRR*, abs/2504.02976.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, et al. 2024. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954.

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. 2023. 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*, pages 2397–2430. PMLR.

Sid Black, Lee Sharkey, Leo Grinsztajn, Eric Winsor, Dan Braun, et al. 2022. Interpreting neural networks through the polytope lens. *Preprint*, arXiv:2211.12312.

Anwoy Chatterjee, H. S. V. N. S. Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2025. On the effect of instruction tuning loss on generalization. *Trans. Assoc. Comput. Linguistics*, 13:1360–1380.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Guy Davidson, Todd M. Gureckis, Brenden M. Lake, and Adina Williams. 2025. Do different prompting methods yield a common task representation in language models? *CoRR*, abs/2505.12075.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Jaden Fried Fiotto-Kaufman, Alexander Russell Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, et al. 2025. Nnsight and NDIF: democratizing access to open-weight foundation model internals. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S., Deepali Aneja, et al. 2024. A closer look at the limitations of instruction tuning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Boris Hanin and David Rolnick. 2019. Deep relu networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, et al. 2025. SAIF: A sparse autoencoder framework for interpreting and steering instruction following of language models. *CoRR*, abs/2502.11356.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Association for Computational Linguistics.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Shirley You Ren, Kwan Ho Ryan Chan, Udhyakumar Nallasamy, Andrew Miller, and Jaya Narain. 2024. Do LLMs internally "know" when they follow instructions? In *MINT: Foundation Model Interventions*.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu-Tung Lin, et al. 2024. Chat vector: A simple approach to equip llms with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand, August 11-16, 2024*. Association for Computational Linguistics.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, et al. 2024. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuxuan Li, Declan Campbell, Stephanie C. Y. Chan, and Andrew Kyle Lampinen. 2025. Just-in-time and distributed task representations in language models. *CoRR*, abs/2509.04466.

Huanxuan Liao, Shizhu He, Yao Xu, Yuanzhe Zhang, Yanchao Hao, et al. 2024. From instance training to instruction learning: Task adapters generation from instructions. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, et al. 2024. Sparse crosscoders for cross-layer features and model diffing. `https://transformer-circuits.pub/2024/crosscoders/index.html`. Anthropic interpretability research note.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Trung Nguyen and Yan Leng. 2025. Toward a flexible framework for linear representation hypothesis using maximum likelihood estimation. *CoRR*, abs/2502.16385.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2025. 2 olmo 2 furious. *CoRR*, abs/2501.00656.

OpenAI. 2025. gpt-5-nano-2025-08-07.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, et al. 2025. Open problems in mechanistic interpretability. *Trans. Mach. Learn. Res.*, 2025.

Zhengxiang Shi, Adam X. Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, et al. 2024. Instruction tuning with loss over instructions. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Denis Sutter, Julian Minder, Thomas Hofmann, and Tiago Pimentel. 2025. The non-linear representation dilemma: Is causal abstraction enough for mechanistic interpretability? *CoRR*, abs/2507.08802.

Naftali Tishby, Fernando C. N. Pereira, and William Bialek. 2000. The information bottleneck method. *CoRR*, physics/0004057.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, et al. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, et al. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2024. From language modeling to instruction following: Understanding the behavior shift in LLMs after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2341–2369, Mexico City, Mexico. Association for Computational Linguistics.

Experiment (1 Task)	Num. Samples	OLMo-2 1B	OLMo-2 7B
Activation Patching	100	4 hrs.	24 hrs.
Path Analysis (k=1)	100	6 hrs.	-
Path Analysis (k=2)	20	36 hrs.	-
Path Analysis (k=3)	15	40 hrs.	-

Table 4: Average runtimes for 1 task in each experiment phase. We observe that the SFT and DPO checkpoints of each model typically have longer runtimes than the base model.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, et al. 2023. LIMA: less is more for alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Hardware Details and Runtimes

In this section, we provide information regarding the computational resources used as well as the experiment runtimes.

We run our activation patching experiments (Section 5) and our path analysis experiments (Section 6) on H100 and A100 GPUs, with RAM capacities of either 40GB or 80GB. We use non-quantized models. For experiments which were not as resource-intensive, such as the linear probe (Section 5.1) and the inference experiments (Section 3), we use an L40 48GB GPU.

The average runtimes for our main experiments are listed in Table 4.

B Full Activation Patching Results

In this section, we present the activation patching heatmaps for all models (Figures 6 - 11), as well as the numerical logit and rank contributions (Tables 9 - 16). For each model, we display the results of the 10 highest-contributing tokens.

As a supplement, in Figures 12 - 19, we additionally present the activation patching heatmaps in their *unnormalized* form - that is, without normalizing against the mininum and maximum scores across all panels in the diagram. While the underlying data is the same as in Figures 6 - 11, this alternate view makes it easier to see the areas of localization in each model.

Finally, we present the t-test results conducted on OLMo 1B and OLMo 7B models (Tables 5 -

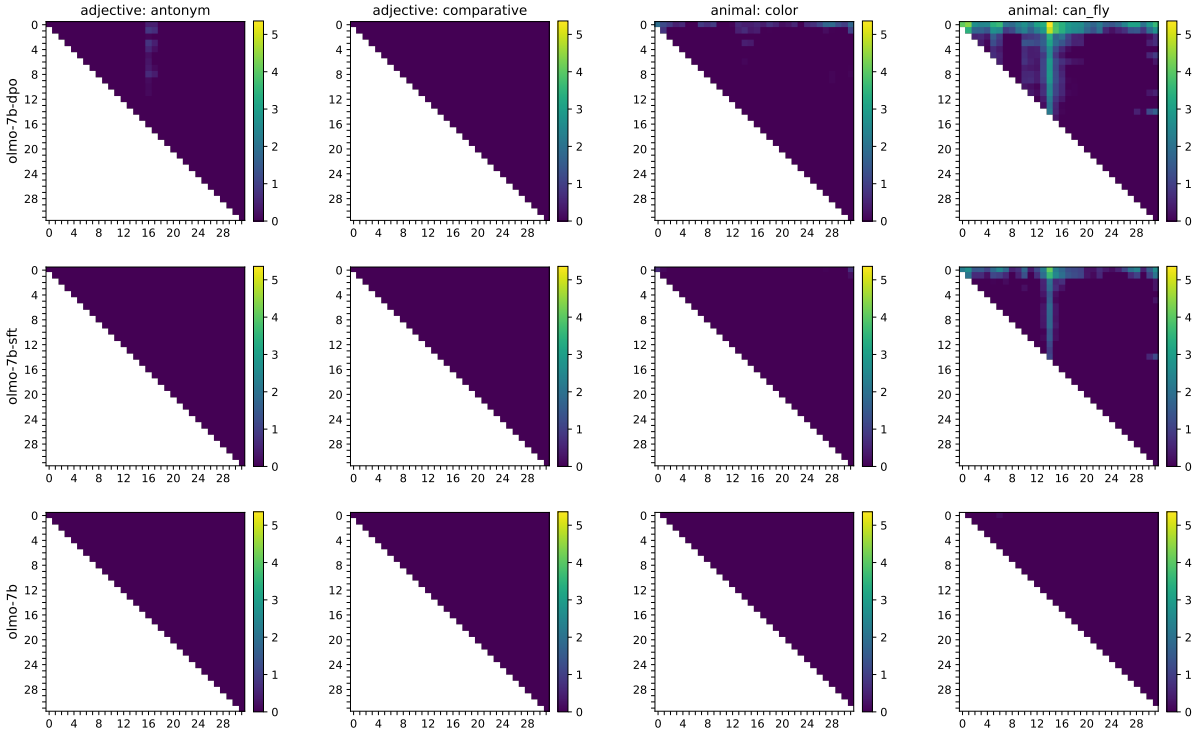

Figure 6: Effects of 1- and 2-layer patching configurations on the logit of the target token for OLMo-2 7B models. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

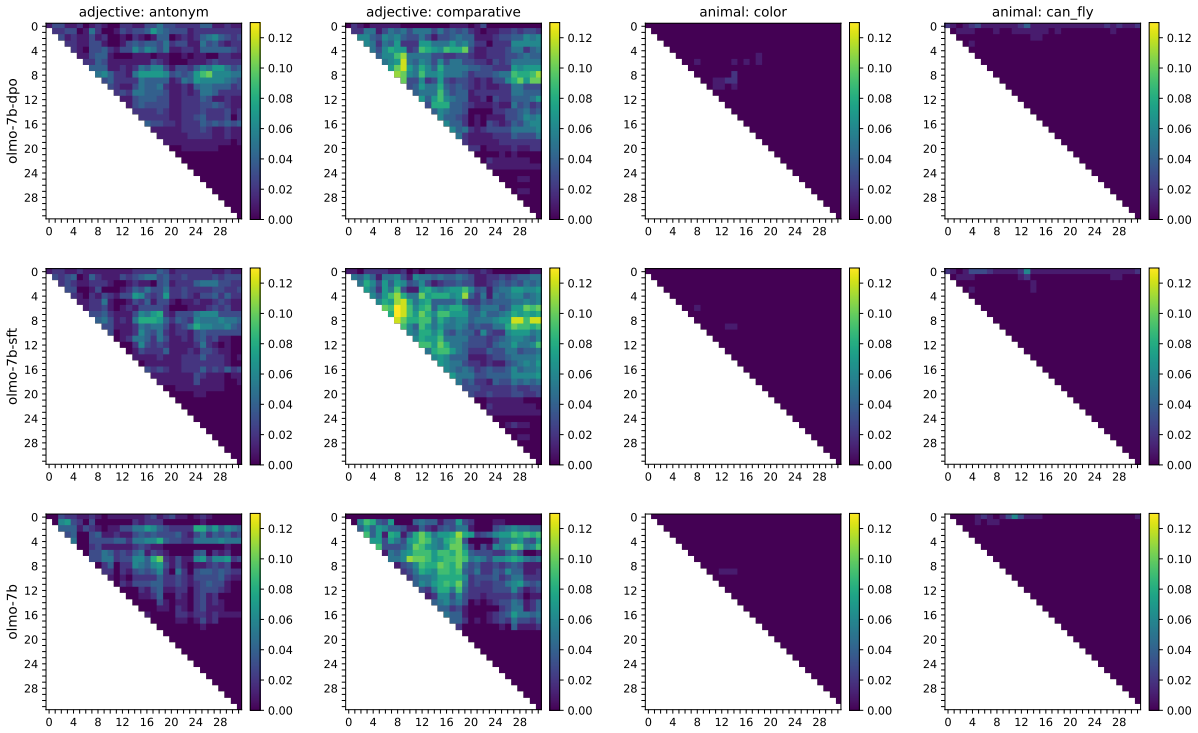

Figure 7: Effects of 1- and 2-layer patching configurations on the rank of the target token for OLMo-2 7B models. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

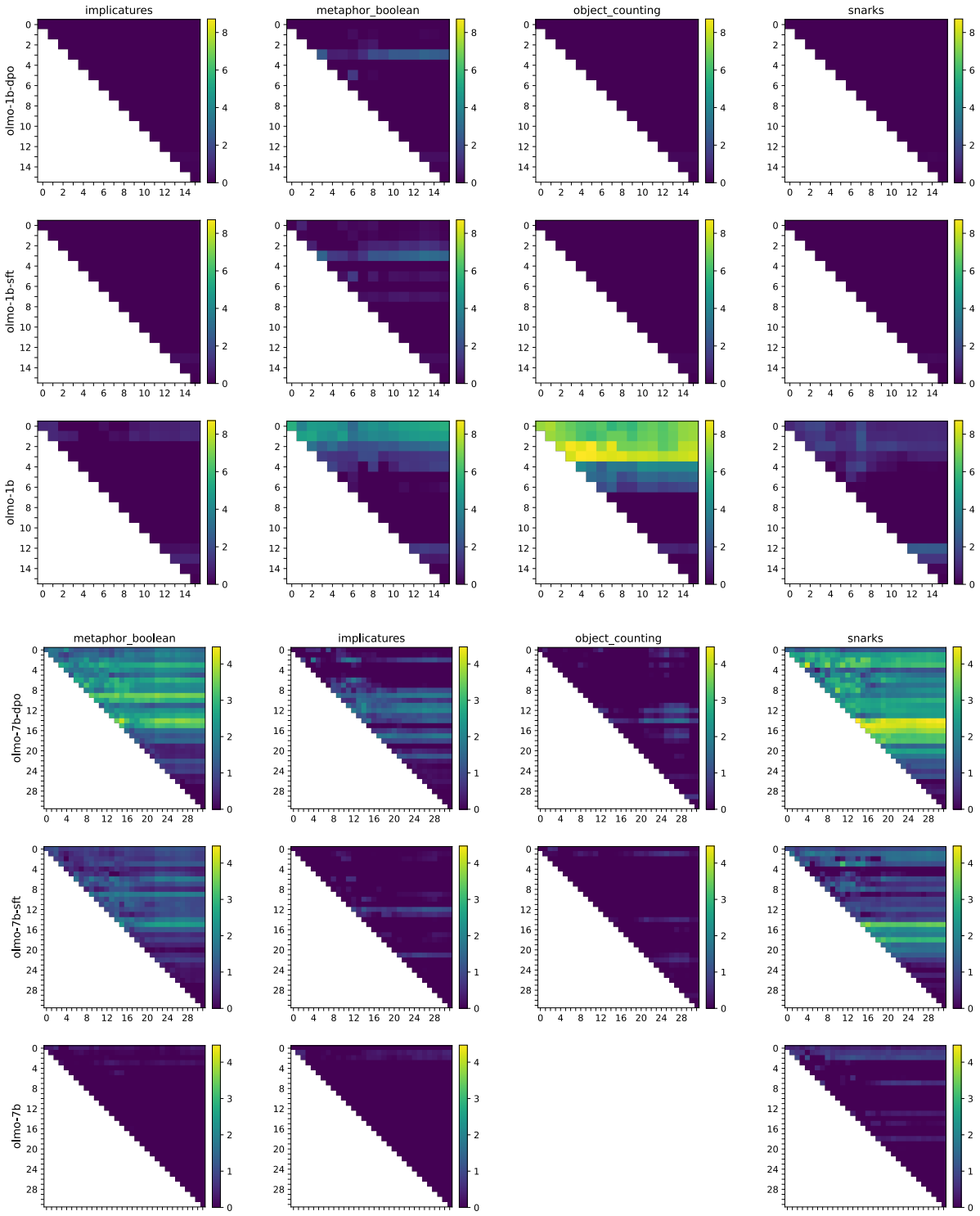

Figure 8: Effects of 1- and 2-layer patching configurations on the logit of the target token for BigBench tasks. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching. OBJECT_COUNTING scores for OLMo-2 7B are missing due to a model error while running the task.

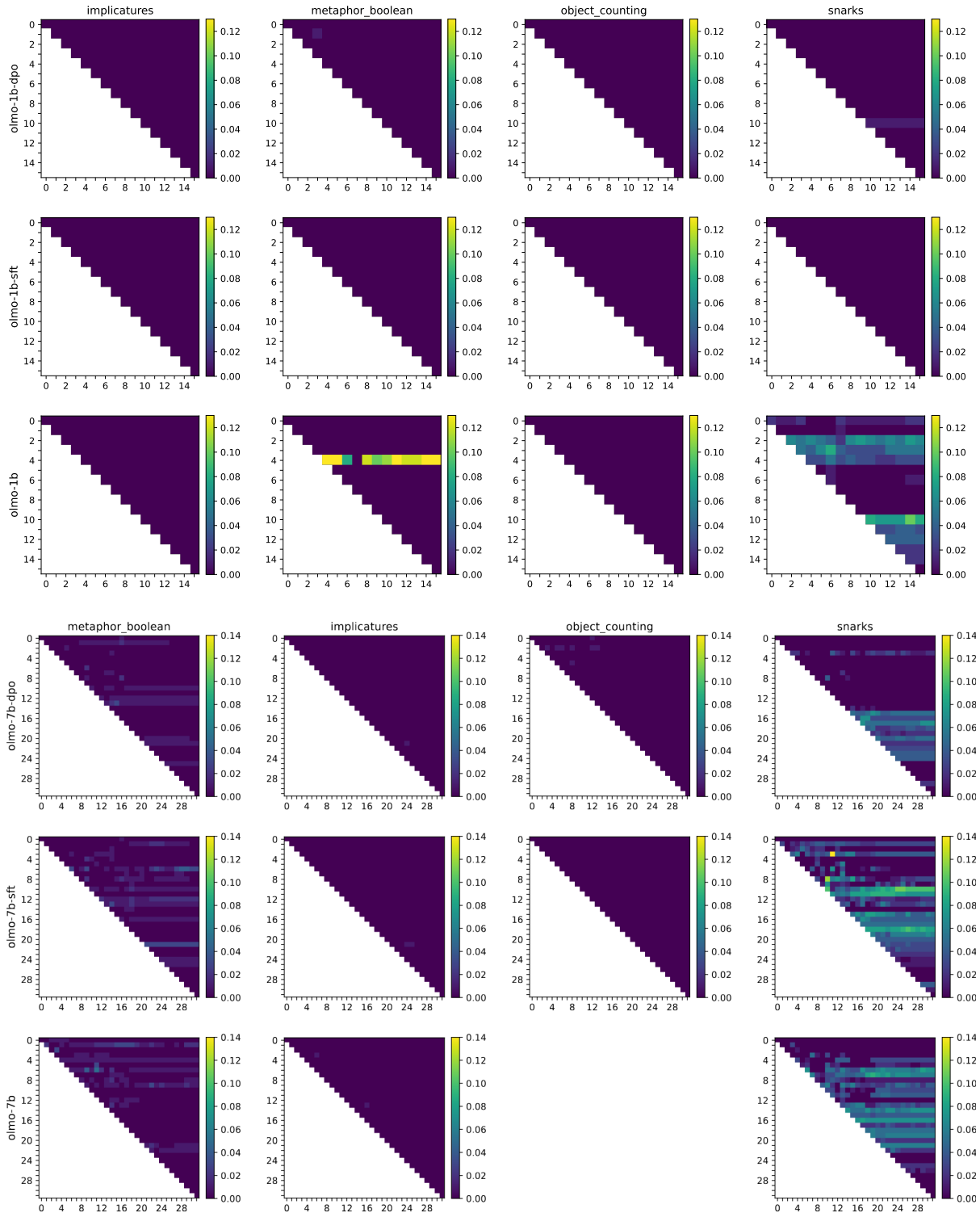

Figure 9: Effects of 1- and 2-layer patching configurations on the rank of the target token for BigBench tasks. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching. OBJECT_COUNTING scores for OLMo-2 7B are missing due to a model error while running the task.

8). For each task, we pick the top-10 patching combinations in terms of average rank contribution (as shown in the heatmaps, as well as the numerical score tables). Then, for each of these layer combinations, we take their unnormalized patching scores and evaluate Eq. 2 (as a boolean truth value 1 or 0).

C Task Inference Results

Some of our tasks are not easily evaluated by EMA metrics due to having multiple valid answer options. For example, the ADJ: ANTONYM task expects the model to output the antonym of an adjective. However, multiple antonyms can exist for a given word, and oftentimes cannot be exhaustively listed. While the correctness of an output can still be measured through similarity metrics, this would not capture instruction following – i.e. whether the model produced an appropriate response to the instruction, even if the answer is wrong.

Thus, for such tasks, we manually define a scope of accepted response types and implement an LLM-as-a-Judge strategy using GPT-5-nano (OpenAI, 2025). We prompt the judge to answer 'yes' or 'no' as to whether or not the response is appropriate given the instruction, regardless of answer correctness. If the judge answers 'yes', we mark the answer as correct. We refer to this metric as INSTRUCTIONAL ACCURACY (IA).

Table 19 shows the prompts provided for the judge for each task, with the manually-defined criteria in bold. We report the EMA and IA scores of our models in Figure 20.

D Vector Space Analysis Results

In Figure 21, we present the LDA task representations for each of our 8 tasks. For each task, we produce 200 rephrasings of the instruction as described in Section 5.1.

We prompt ChatGPT-5-nano (OpenAI, 2025) using the following prompt:

> This is a task instruction: '{}' Please rephrase this instruction in 200 different ways without changing the essential meaning and without specifying any particular statements the speakers made. Please vary the rephrasals in terms of syntax, linguistic style, and complexity.

where we fill in the blank using the standard instruction in Table 1.

E Path Analysis Results with top-k Attention

In this section, we present supplementary results for our path analysis experiments in Section 6. These results reinforce the role of the T_{inst} as a primary source of high-ranking paths, particularly in comparison to later, non-instructional tokens.

Figures 22 - 24 show the fraction of high-ranking paths contributed by each token position, across all examined tokens (positions 7 - 16). Due to long computation times and memory requirements, we do not analyze the full prompt. We notice that the T_{inst} token tends to yield the greatest portion of high-ranking paths across model checkpoints for ADJECTIVES: COMPARATIVE, and consistently more high-ranking paths than later tokens. For ADJECTIVES: ANTONYM, $T_{\text{inst}} - 1$ (Token 7) yields more high-ranking paths for all models in $k{>}1$[4]. This could be due to the greater open-endedness of the antonyms task, resulting in a wider pool of possible answers that must be sorted through by the model.

F Attention Head Activity Experiments

In this section, we present the full details of our experiments regarding attention head activity.

F.1 Confidence and Significance Tests

Figures 25 - 30 show the mean activity rates for attention heads across the adjectives tasks, along with the bootstrapped confidence intervals and p-values.

F.2 Similarity of Head Activity Across Contrastive Instructions and k Values

k Values. To determine the impact of top-k attention with various values of k on the head activity patterns, we calculate the Jaccard similarity of the average head activity between $k{=}1$ and $k{=}3$ (Table 20), as well as between $k{=}2$ and $k{=}3$ (Table 21). The comparison between $k{=}1$ and $k{=}2$ is given in Table 3.

We find that from $k{=}1$ to $k{=}2$, there is a moderate similarity in head activity, mostly in the early layers (Table 20). The lower similarity in the later layers indicates that there is some change in head activity here. However, when going from $k{=}2$ to $k{=}3$, the similarity scores are higher overall – this indicates that the head activity pattern has mostly

[4]We note that, due to experiment constraints, we were unable to run the SFT model in the $k{=}3$ setting.

Contrastive Tasks: T-Test											
OLMo-2 1B SFT											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(3, 3)	-1.003e+03	3.487e-200	(0, 0)	-1.132e+03	2.327e-205	(3, 13)	-1.334e+03	1.941e-212	(3, 7)	-1.012e+03	1.483e-200
(3, 14)	-1.504e+03	1.377e-217	(0, 15)	-inf	0.000e+00	(5, 13)	-9.890e+02	1.447e-199	(1, 7)	-9.862e+02	1.911e-199
(3, 15)	-2.239e+03	1.061e-234	(0, 14)	-2.490e+03	2.901e-239	(3, 12)	-1.504e+03	1.377e-217	(2, 7)	-1.036e+03	1.498e-201
(0, 0)	-1.182e+03	3.077e-207	(4, 6)	-9.963e+02	6.999e-200	(5, 12)	-1.081e+03	2.208e-203	(3, 4)	-9.934e+02	9.296e-200
(0, 15)	-inf	0.000e+00	(2, 3)	-1.117e+03	8.176e-205	(0, 13)	-1.562e+03	3.312e-219	(3, 8)	-9.890e+02	1.446e-199
(3, 11)	-1.092e+03	7.928e-204	(3, 3)	-1.036e+03	1.412e-201	(2, 13)	-2.055e+03	5.115e-231	(3, 9)	-9.874e+02	1.696e-199
(3, 13)	-1.081e+03	2.208e-203	(3, 14)	-2.239e+03	1.061e-234	(3, 11)	-1.504e+03	1.377e-217	(1, 6)	-1.007e+03	2.337e-200
(0, 14)	-1.409e+03	8.728e-215	(3, 15)	-2.490e+03	2.901e-239	(3, 14)	-2.055e+03	5.115e-231	(3, 5)	-9.890e+02	1.446e-199
(2, 3)	-1.248e+03	1.493e-209	(2, 5)	-1.164e+03	1.437e-206	(2, 12)	-2.490e+03	2.901e-239	(3, 6)	-1.007e+03	2.337e-200
(0, 13)	-1.117e+03	8.176e-205	(2, 6)	-1.182e+03	3.077e-207	(6, 12)	-1.104e+03	2.646e-204	(3, 11)	-1.334e+03	1.941e-212
OLMo-2 1B DPO											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(3, 14)	-1.504e+03	1.377e-217	(0, 0)	-1.092e+03	7.928e-204	(0, 13)	-1.800e+03	2.590e-225	(3, 7)	-9.850e+02	2.153e-199
(3, 3)	-9.934e+02	9.296e-200	(0, 14)	-2.055e+03	5.115e-231	(5, 13)	-9.850e+02	2.153e-199	(1, 7)	-9.874e+02	1.696e-199
(3, 13)	-1.092e+03	7.928e-204	(0, 15)	-inf	0.000e+00	(5, 12)	-1.104e+03	2.646e-204	(3, 4)	-1.061e+03	1.400e-202
(3, 15)	-1.800e+03	2.590e-225	(2, 3)	-1.061e+03	1.400e-202	(3, 12)	-1.504e+03	1.377e-217	(4, 7)	-1.036e+03	1.498e-201
(0, 0)	-1.117e+03	8.176e-205	(3, 14)	-2.239e+03	1.061e-234	(3, 13)	-1.504e+03	1.377e-217	(1, 6)	-1.003e+03	3.523e-200
(0, 15)	-inf	0.000e+00	(5, 6)	-9.854e+02	2.070e-199	(3, 11)	-1.504e+03	1.377e-217	(3, 5)	-9.862e+02	1.912e-199
(0, 14)	-1.504e+03	1.377e-217	(2, 2)	-1.092e+03	7.928e-204	(7, 13)	-1.029e+03	2.730e-201	(3, 6)	-9.874e+02	1.696e-199
(3, 11)	-1.061e+03	1.400e-202	(2, 6)	-1.061e+03	1.400e-202	(2, 13)	-3.484e+03	1.053e-253	(3, 8)	-9.862e+02	1.912e-199
(0, 3)	-1.044e+03	6.922e-202	(2, 13)	-1.707e+03	5.030e-223	(5, 11)	-1.008e+03	2.303e-200	(0, 5)	-1.389e+03	3.596e-214
(0, 13)	-1.117e+03	8.176e-205	(2, 14)	-1.914e+03	6.080e-228	(5, 14)	-9.874e+02	1.696e-199	(0, 7)	-1.389e+03	3.596e-214
OLMo-2 1B											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(1, 14)	-1.224e+03	1.002e-208	(1, 1)	-1.334e+03	1.941e-212	(6, 13)	-9.874e+02	1.697e-199	(4, 8)	-9.934e+02	9.296e-200
(1, 1)	-1.202e+03	5.898e-208	(1, 14)	-1.274e+03	1.925e-210	(6, 7)	-9.934e+02	9.313e-200	(3, 8)	-1.092e+03	7.928e-204
(1, 15)	-inf	0.000e+00	(1, 15)	-inf	0.000e+00	(6, 14)	-1.334e+03	1.941e-212	(5, 8)	-1.504e+03	1.377e-217
(0, 1)	-1.707e+03	5.030e-223	(6, 14)	-1.274e+03	1.925e-210	(7, 14)	-1.147e+03	6.063e-206	(5, 9)	-1.707e+03	5.030e-223
(1, 13)	-1.036e+03	1.412e-201	(7, 9)	-1.224e+03	1.002e-208	(6, 6)	-1.069e+03	6.546e-203	(3, 10)	-1.248e+03	1.493e-209
(1, 6)	-1.104e+03	2.646e-204	(6, 6)	-1.081e+03	2.208e-203	(6, 8)	-9.850e+02	2.153e-199	(4, 9)	-1.070e+03	5.744e-203
(3, 9)	-1.081e+03	2.208e-203	(6, 15)	-inf	0.000e+00	(6, 12)	-1.023e+03	5.009e-201	(3, 9)	-1.274e+03	1.925e-210
(6, 8)	-1.224e+03	1.002e-208	(1, 13)	-1.023e+03	5.009e-201	(6, 15)	-4.901e+03	2.232e-268	(4, 10)	-1.070e+03	5.744e-203
(6, 9)	-1.248e+03	1.493e-209	(7, 10)	-1.202e+03	5.898e-208	(6, 11)	-1.070e+03	5.744e-203	(4, 7)	-1.117e+03	8.176e-205
(7, 9)	-1.454e+03	4.028e-216	(0, 1)	-1.409e+03	8.728e-215	(7, 7)	-9.995e+02	5.083e-200	(4, 11)	-1.147e+03	6.063e-206

Table 5: Contrastive tasks T-Test results for OLMo-2 1B models.

BigBench Tasks: T-Test											
OLMo-2 1B SFT											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(3, 13)	-1.029e+03	2.730e-201	(13, 14)	-1.831e+03	4.827e-226	(13, 13)	-1.177e+03	4.881e-207	(13, 14)	-1.104e+03	2.646e-204
(3, 14)	-1.409e+03	8.728e-215	(13, 13)	-1.430e+03	2.068e-215	(13, 14)	-inf	0.000e+00	(13, 13)	-1.353e+03	4.982e-213
(3, 3)	-1.290e+03	5.353e-211	(13, 15)	-inf	0.000e+00	(13, 15)	-inf	0.000e+00	(13, 15)	-inf	0.000e+00
(3, 15)	-inf	0.000e+00	(14, 14)	-4.999e+03	3.144e-269	(14, 14)	-inf	0.000e+00	(15, 15)	-inf	0.000e+00
(3, 12)	-1.202e+03	5.898e-208	(14, 15)	-inf	0.000e+00	(14, 15)	-inf	0.000e+00	(14, 14)	-1.104e+03	2.646e-204
(3, 11)	-9.963e+02	6.999e-200	(15, 15)	-inf	0.000e+00	(15, 15)	-inf	0.000e+00	(14, 15)	-inf	0.000e+00
(3, 10)	-3.484e+03	1.053e-253	(12, 14)	-1.061e+03	1.400e-202	(12, 13)	-1.069e+03	6.546e-203	(12, 13)	-1.143e+03	8.560e-206
(3, 8)	-1.052e+03	3.206e-202	(12, 13)	-1.263e+03	4.322e-210	(12, 14)	-4.999e+03	3.144e-269	(12, 14)	-1.104e+03	2.646e-204
(3, 9)	-2.055e+03	5.115e-231	(12, 12)	-3.484e+03	1.053e-253	(12, 12)	-1.628e+03	5.250e-221	(12, 12)	-inf	0.000e+00
(3, 4)	-1.090e+03	9.535e-204	(12, 15)	-inf	0.000e+00	(12, 15)	-inf	0.000e+00	(12, 15)	-inf	0.000e+00
OLMo-2 1B DPO											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(3, 13)	-9.890e+02	1.446e-199	(13, 14)	-1.947e+03	1.089e-228	(13, 13)	-1.216e+03	1.849e-208	(13, 14)	-1.029e+03	2.730e-201
(3, 14)	-1.224e+03	1.002e-208	(13, 13)	-1.043e+03	7.448e-202	(13, 14)	-inf	0.000e+00	(13, 13)	-1.069e+03	6.546e-203
(3, 10)	-2.055e+03	5.115e-231	(13, 15)	-inf	0.000e+00	(13, 15)	-inf	0.000e+00	(13, 15)	-inf	0.000e+00
(3, 3)	-1.177e+03	4.881e-207	(14, 14)	-4.999e+03	3.144e-269	(14, 14)	-inf	0.000e+00	(15, 15)	-inf	0.000e+00
(3, 15)	-inf	0.000e+00	(14, 15)	-inf	0.000e+00	(14, 15)	-inf	0.000e+00	(14, 14)	-1.017e+03	8.739e-201
(3, 11)	-1.061e+03	1.400e-202	(15, 15)	-inf	0.000e+00	(15, 15)	-inf	0.000e+00	(14, 15)	-inf	0.000e+00
(3, 12)	-1.454e+03	4.028e-216	(12, 14)	-1.147e+03	6.063e-206	(12, 13)	-1.102e+03	3.288e-204	(12, 14)	-1.023e+03	5.009e-201
(3, 8)	-9.963e+02	6.999e-200	(12, 13)	-1.012e+03	1.483e-200	(12, 14)	-2.280e+03	1.754e-235	(12, 12)	-1.454e+03	4.028e-216
(3, 9)	-1.454e+03	4.028e-216	(12, 12)	-inf	0.000e+00	(12, 12)	-1.914e+03	6.080e-228	(12, 15)	-inf	0.000e+00
(5, 6)	-inf	0.000e+00	(12, 15)	-inf	0.000e+00	(12, 15)	-inf	0.000e+00	(12, 13)	-1.060e+03	1.561e-202
OLMo-2 1B											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(0, 0)	-4.999e+03	3.144e-269	(1, 14)	-1.475e+03	9.170e-217	(2, 5)	-1.947e+03	1.089e-228	(12, 13)	-inf	0.000e+00
(0, 15)	-inf	0.000e+00	(0, 1)	-1.060e+03	1.561e-202	(3, 4)	-4.999e+03	3.144e-269	(12, 14)	-inf	0.000e+00
(0, 14)	-1.947e+03	1.089e-228	(1, 1)	-1.216e+03	1.849e-208	(3, 5)	-1.947e+03	1.089e-228	(12, 12)	-3.552e+03	1.544e-254
(0, 13)	-inf	0.000e+00	(1, 15)	-inf	0.000e+00	(2, 3)	-inf	0.000e+00	(12, 15)	-inf	0.000e+00
(1, 2)	-1.655e+03	1.060e-221	(0, 14)	-2.537e+03	4.608e-240	(3, 6)	-1.353e+03	4.982e-213	(1, 7)	-2.490e+03	2.901e-239
(0, 12)	-3.552e+03	1.544e-254	(0, 0)	-1.060e+03	1.561e-202	(2, 7)	-1.029e+03	2.730e-201	(2, 7)	-1.707e+03	5.030e-223
(0, 11)	-1.224e+03	1.002e-208	(0, 15)	-inf	0.000e+00	(3, 8)	-1.800e+03	2.590e-225	(0, 3)	-9.890e+02	1.447e-199
(1, 1)	-3.552e+03	1.544e-254	(1, 11)	-inf	0.000e+00	(2, 4)	-3.552e+03	1.544e-254	(0, 7)	-1.800e+03	2.590e-225
(1, 15)	-inf	0.000e+00	(13, 14)	-inf	0.000e+00	(3, 7)	-9.890e+02	1.446e-199	(2, 4)	-1.003e+03	3.487e-200
(1, 13)	-4.999e+03	3.144e-269	(1, 9)	-inf	0.000e+00	(3, 3)	-inf	0.000e+00	(4, 7)	-2.860e+03	3.234e-245

Table 6: BigBench tasks T-Test results for OLMo-2 1B models.

Contrastive Tasks: T-Test											
OLMo-2 7B SFT											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(7, 18)	-1.370e+03	1.458e-213	(6, 8)	-1.044e+03	6.922e-202	(6, 8)	-9.910e+02	1.184e-199	(0, 13)	-9.963e+02	6.999e-200
(8, 15)	-1.182e+03	3.077e-207	(7, 8)	-1.061e+03	1.400e-202	(9, 13)	-1.061e+03	1.400e-202	(0, 4)	-9.850e+02	2.153e-199
(8, 16)	-1.202e+03	5.898e-208	(5, 9)	-1.092e+03	7.928e-204	(9, 14)	-1.117e+03	8.176e-205	(0, 5)	-9.910e+02	1.184e-199
(8, 17)	-1.274e+03	1.925e-210	(6, 9)	-1.061e+03	1.400e-202	(2, 16)	-1.008e+03	2.303e-200	(0, 6)	-1.061e+03	1.400e-202
(9, 24)	-2.490e+03	2.901e-239	(7, 9)	-1.070e+03	5.744e-203	(2, 17)	-1.070e+03	5.744e-203	(0, 0)	-1.128e+03	3.131e-205
(4, 14)	-1.202e+03	5.898e-208	(8, 8)	-1.036e+03	1.412e-201	(3, 15)	-1.164e+03	1.437e-206	(0, 7)	-9.996e+02	5.049e-200
(4, 19)	-1.454e+03	4.028e-216	(8, 27)	-2.490e+03	2.901e-239	(4, 13)	-9.996e+02	5.049e-200	(0, 12)	-9.862e+02	1.912e-199
(7, 17)	-1.248e+03	1.493e-209	(8, 28)	-inf	0.000e+00	(4, 14)	-1.008e+03	2.303e-200	(0, 31)	-3.484e+03	1.053e-253
(8, 18)	-1.504e+03	1.377e-217	(8, 30)	-4.901e+03	2.232e-268	(5, 16)	-1.081e+03	2.208e-203	(1, 13)	-9.890e+02	1.447e-199
(4, 15)	-1.202e+03	5.898e-208	(8, 31)	-inf	0.000e+00	(6, 6)	-1.003e+03	3.487e-200	(0, 1)	-1.043e+03	7.448e-202
OLMo-2 7B DPO											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(8, 26)	-inf	0.000e+00	(6, 9)	-1.061e+03	1.400e-202	(8, 14)	-9.850e+02	2.153e-199	(0, 13)	-1.023e+03	5.196e-201
(8, 25)	-4.901e+03	2.232e-268	(7, 9)	-1.036e+03	1.412e-201	(9, 13)	-9.934e+02	9.296e-200	(0, 5)	-1.060e+03	1.561e-202
(1, 17)	-1.274e+03	1.925e-210	(5, 9)	-1.117e+03	8.176e-205	(9, 14)	-9.910e+02	1.184e-199	(0, 0)	-1.655e+03	1.060e-221
(7, 27)	-2.490e+03	2.901e-239	(8, 8)	-1.036e+03	1.412e-201	(5, 18)	-1.052e+03	3.206e-202	(0, 4)	-1.060e+03	1.561e-202
(7, 28)	-inf	0.000e+00	(8, 31)	-inf	0.000e+00	(6, 8)	-9.862e+02	1.912e-199	(0, 12)	-1.017e+03	8.988e-201
(8, 15)	-1.132e+03	2.327e-205	(4, 19)	-1.454e+03	4.028e-216	(6, 14)	-1.012e+03	1.452e-200	(0, 26)	-1.104e+03	2.646e-204
(8, 16)	-1.164e+03	1.437e-206	(7, 8)	-1.044e+03	6.922e-202	(6, 16)	-9.934e+02	9.296e-200	(0, 28)	-1.334e+03	1.941e-212
(8, 17)	-1.224e+03	1.002e-208	(8, 9)	-1.036e+03	1.412e-201	(6, 18)	-1.052e+03	3.206e-202	(0, 31)	-1.454e+03	4.028e-216
(8, 18)	-1.370e+03	1.458e-213	(9, 9)	-1.044e+03	6.922e-202	(9, 11)	-9.962e+02	7.026e-200	(0, 1)	-1.290e+03	5.353e-211
(8, 24)	-2.860e+03	3.234e-245	(9, 28)	-4.901e+03	2.232e-268	(9, 12)	-9.890e+02	1.447e-199	(0, 3)	-1.389e+03	3.596e-214
OLMo-2 7B											
	adj: antonym			adj: comparative			anim: color			anim: can_fly	
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(7, 18)	-2.490e+03	2.901e-239	(3, 18)	-2.490e+03	2.901e-239	(9, 12)	-inf	0.000e+00	(0, 11)	-inf	0.000e+00
(2, 24)	-4.901e+03	2.232e-268	(7, 10)	-2.055e+03	5.115e-231	(9, 13)	-inf	0.000e+00	(0, 10)	-inf	0.000e+00
(2, 25)	-inf	0.000e+00	(3, 12)	-2.055e+03	5.115e-231	(9, 14)	-inf	0.000e+00	(0, 12)	-inf	0.000e+00
(4, 18)	-2.239e+03	1.061e-234	(4, 12)	-1.800e+03	2.590e-225	(2, 16)	-1.914e+03	6.080e-228	(0, 6)	-3.484e+03	1.053e-253
(7, 17)	-2.239e+03	1.061e-234	(4, 18)	-1.914e+03	6.080e-228	(4, 5)	-inf	0.000e+00	(0, 0)	-1.164e+03	1.437e-206
(1, 3)	-1.628e+03	5.250e-221	(5, 11)	-1.504e+03	1.377e-217	(5, 5)	-2.490e+03	2.901e-239	(0, 5)	-2.055e+03	5.115e-231
(2, 16)	-1.202e+03	5.898e-208	(5, 17)	-1.628e+03	5.250e-221	(5, 12)	-4.901e+03	2.232e-268	(0, 7)	-2.490e+03	2.901e-239
(2, 27)	-4.901e+03	2.232e-268	(5, 18)	-1.562e+03	3.312e-219	(5, 28)	-4.901e+03	2.232e-268	(0, 9)	-4.901e+03	2.232e-268
(7, 25)	-inf	0.000e+00	(5, 27)	-3.484e+03	1.053e-253	(5, 31)	-inf	0.000e+00	(0, 13)	-inf	0.000e+00
(7, 29)	-inf	0.000e+00	(6, 18)	-2.239e+03	1.061e-234	(6, 6)	-4.901e+03	2.232e-268	(0, 14)	-inf	0.000e+00

Table 7: Contrastive tasks T-Test results for OLMo-2 7B models.

BigBench Tasks: T-Test											
OLMo-2 7B SFT											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(15, 24)	-9.850e+02	2.153e-199	(12, 26)	-1.061e+03	1.400e-202	(1, 25)	-1.562e+03	3.312e-219	(15, 16)	-1.430e+03	2.068e-215
(15, 22)	-1.029e+03	2.861e-201	(12, 14)	-1.044e+03	6.922e-202	(1, 24)	-1.800e+03	2.590e-225	(15, 28)	-1.012e+03	1.483e-200
(15, 27)	-1.070e+03	5.744e-203	(12, 27)	-2.055e+03	5.115e-231	(0, 2)	-2.239e+03	1.061e-234	(15, 30)	-1.302e+03	2.114e-211
(15, 23)	-9.850e+02	2.153e-199	(12, 28)	-1.070e+03	5.744e-203	(14, 27)	-inf	0.000e+00	(15, 15)	-1.947e+03	1.089e-228
(15, 26)	-9.862e+02	1.911e-199	(12, 29)	-4.901e+03	2.232e-268	(22, 27)	-inf	0.000e+00	(15, 31)	-inf	0.000e+00
(15, 28)	-1.117e+03	8.176e-205	(12, 30)	-2.860e+03	3.234e-245	(1, 27)	-inf	0.000e+00	(15, 29)	-9.962e+02	7.026e-200
(15, 21)	-9.995e+02	5.083e-200	(12, 25)	-1.117e+03	8.176e-205	(1, 23)	-1.104e+03	2.646e-204	(15, 18)	-2.537e+03	4.608e-240
(15, 29)	-inf	0.000e+00	(12, 12)	-1.239e+03	3.019e-209	(1, 28)	-3.484e+03	1.053e-253	(15, 26)	-1.182e+03	3.077e-207
(15, 25)	-1.003e+03	3.523e-200	(12, 24)	-1.036e+03	1.412e-201	(22, 26)	-4.901e+03	2.232e-268	(15, 19)	-1.586e+03	6.960e-220
(15, 30)	-2.239e+03	1.061e-234	(12, 31)	-inf	0.000e+00	(14, 21)	-1.029e+03	2.730e-201	(15, 27)	-1.051e+03	3.507e-202
OLMo-2 7B DPO											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(14, 15)	-1.947e+03	1.089e-228	(12, 26)	-9.874e+02	1.696e-199	(14, 27)	-inf	0.000e+00	(14, 30)	-1.036e+03	1.498e-201
(14, 27)	-1.370e+03	1.458e-213	(12, 14)	-1.159e+03	2.143e-206	(14, 28)	-inf	0.000e+00	(14, 14)	-1.947e+03	1.089e-228
(14, 26)	-9.995e+02	5.083e-200	(11, 14)	-1.079e+03	2.581e-203	(14, 29)	-9.854e+02	2.069e-199	(14, 31)	-inf	0.000e+00
(9, 22)	-1.090e+03	9.535e-204	(12, 27)	-1.008e+03	2.303e-200	(14, 25)	-1.182e+03	3.077e-207	(14, 28)	-1.389e+03	3.596e-214
(14, 22)	-1.090e+03	9.535e-204	(12, 29)	-1.628e+03	5.250e-221	(14, 26)	-1.800e+03	2.590e-225	(14, 29)	-1.003e+03	3.523e-200
(14, 24)	-1.102e+03	3.288e-204	(12, 30)	-1.202e+03	5.898e-208	(14, 21)	-1.504e+03	1.377e-217	(15, 16)	-2.537e+03	4.608e-240
(14, 28)	-1.224e+03	1.002e-208	(12, 24)	-9.850e+02	2.153e-199	(12, 28)	-2.860e+03	3.234e-245	(14, 27)	-1.069e+03	6.546e-203
(14, 23)	-1.060e+03	1.561e-202	(12, 28)	-9.854e+02	2.070e-199	(12, 27)	-4.901e+03	2.232e-268	(14, 19)	-1.389e+03	3.596e-214
(9, 27)	-1.248e+03	1.493e-209	(12, 25)	-1.012e+03	1.483e-200	(14, 22)	-2.055e+03	5.115e-231	(14, 26)	-1.007e+03	2.337e-200
(9, 14)	-2.915e+03	4.935e-246	(12, 21)	-1.430e+03	2.068e-215	(12, 29)	-1.430e+03	2.068e-215	(15, 17)	-1.947e+03	1.089e-228
OLMo-2 7B											
metaphor_boolean			implicatures			object_counting			snarks		
Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value	Layer(s)	t-stat.	p-value
(0, 2)	-9.934e+02	9.296e-200	(1, 2)	-1.090e+03	9.535e-204	(14, 27)	-	-	(2, 13)	-9.874e+02	1.697e-199
(0, 29)	-3.484e+03	1.053e-253	(1, 1)	-1.115e+03	1.055e-204	(14, 28)	-	-	(2, 4)	-9.910e+02	1.184e-199
(0, 10)	-1.081e+03	2.208e-203	(1, 31)	-inf	0.000e+00	(14, 29)	-	-	(2, 25)	-1.914e+03	6.080e-228
(0, 6)	-1.202e+03	5.898e-208	(1, 30)	-inf	0.000e+00	(14, 25)	-	-	(2, 14)	-1.070e+03	5.744e-203
(0, 7)	-9.890e+02	1.447e-199	(1, 27)	-inf	0.000e+00	(14, 26)	-	-	(2, 12)	-1.132e+03	2.327e-205
(0, 8)	-9.995e+02	5.083e-200	(1, 25)	-1.504e+03	1.377e-217	(14, 21)	-	-	(2, 23)	-4.901e+03	2.232e-268
(3, 13)	-1.628e+03	5.250e-221	(2, 25)	-2.239e+03	1.061e-234	(12, 28)	-	-	(2, 24)	-4.901e+03	2.232e-268
(0, 5)	-9.910e+02	1.184e-199	(2, 2)	-1.102e+03	3.288e-204	(12, 27)	-	-	(2, 26)	-1.914e+03	6.080e-228
(0, 30)	-2.490e+03	2.901e-239	(2, 31)	-inf	0.000e+00	(14, 22)	-	-	(2, 15)	-9.890e+02	1.446e-199
(0, 1)	-9.934e+02	9.313e-200	(1, 17)	-inf	0.000e+00	(12, 29)	-	-	(2, 8)	-1.008e+03	2.303e-200

Table 8: BigBench tasks T-Test results for OLMo-2 7B models. OBJECT_COUNTING scores for OLMo-2 7B are missing due to a model error while running the task.

Contrastive Tasks - Rank 1B							
OLMo-2 1B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(3, 3)	0.14	(0, 0)	0.08	(3, 13)	0.11	(3, 7)	0.05
(3, 14)	0.14	(0, 15)	0.08	(5, 13)	0.11	(1, 7)	0.03
(3, 15)	0.14	(0, 14)	0.07	(3, 12)	0.10	(2, 7)	0.03
(0, 0)	0.13	(4, 6)	0.07	(5, 12)	0.10	(3, 4)	0.03
(0, 15)	0.13	(2, 3)	0.06	(0, 13)	0.09	(3, 8)	0.03
(3, 11)	0.11	(3, 3)	0.06	(2, 13)	0.09	(3, 9)	0.03
(3, 13)	0.11	(3, 14)	0.06	(3, 11)	0.09	(1, 6)	0.02
(0, 14)	0.10	(3, 15)	0.06	(3, 14)	0.09	(3, 5)	0.02
(2, 3)	0.10	(2, 5)	0.05	(2, 12)	0.08	(3, 6)	0.02
(0, 13)	0.08	(2, 6)	0.05	(6, 12)	0.08	(3, 11)	0.02
OLMo-2 1B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(3, 14)	0.13	(0, 0)	0.07	(0, 13)	0.12	(3, 7)	0.04
(3, 3)	0.11	(0, 14)	0.07	(5, 13)	0.12	(1, 7)	0.03
(3, 13)	0.11	(0, 15)	0.07	(5, 12)	0.11	(3, 4)	0.03
(3, 15)	0.11	(2, 3)	0.03	(3, 12)	0.09	(4, 7)	0.03
(0, 0)	0.10	(3, 14)	0.03	(3, 13)	0.09	(1, 6)	0.02
(0, 15)	0.10	(5, 6)	0.03	(3, 11)	0.08	(3, 5)	0.02
(0, 14)	0.09	(2, 2)	0.02	(7, 13)	0.08	(3, 6)	0.02
(3, 11)	0.08	(2, 6)	0.02	(2, 13)	0.07	(3, 8)	0.02
(0, 3)	0.07	(2, 13)	0.02	(5, 11)	0.07	(0, 5)	0.01
(0, 13)	0.07	(2, 14)	0.02	(5, 14)	0.07	(0, 7)	0.01
OLMo-2 1B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(1, 14)	0.16	(1, 1)	0.08	(6, 13)	0.15	(4, 8)	0.17
(1, 1)	0.14	(1, 14)	0.08	(6, 7)	0.14	(3, 8)	0.15
(1, 15)	0.14	(1, 15)	0.08	(6, 14)	0.13	(5, 8)	0.13
(0, 1)	0.13	(6, 14)	0.07	(7, 14)	0.12	(5, 9)	0.11
(1, 13)	0.11	(7, 9)	0.07	(6, 6)	0.11	(3, 10)	0.08
(1, 6)	0.10	(6, 6)	0.06	(6, 8)	0.11	(4, 9)	0.08
(3, 9)	0.09	(6, 15)	0.06	(6, 12)	0.11	(3, 9)	0.07
(6, 8)	0.09	(1, 13)	0.05	(6, 15)	0.11	(4, 10)	0.07
(6, 9)	0.09	(7, 10)	0.05	(6, 11)	0.10	(4, 7)	0.06
(7, 9)	0.09	(0, 1)	0.04	(7, 7)	0.10	(4, 11)	0.05

Table 9: Numerical rank contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

BigBench Tasks - Logit 1B							
OLMo-2 1B SFT							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(3, 13)	3.16	(13, 14)	0.31	(13, 13)	0.17	(13, 14)	0.21
(3, 14)	3.02	(13, 13)	0.29	(13, 14)	0.17	(13, 13)	0.19
(3, 3)	2.8	(13, 15)	0.29	(13, 15)	0.17	(13, 15)	0.19
(3, 15)	2.8	(14, 14)	0.1	(14, 14)	0.13	(15, 15)	0.0
(3, 12)	2.79	(14, 15)	0.1	(14, 15)	0.13	(14, 14)	-0.05
(3, 11)	2.67	(15, 15)	0.0	(15, 15)	0.0	(14, 15)	-0.05
(3, 10)	2.45	(12, 14)	-0.37	(12, 13)	-0.34	(12, 13)	-0.7
(3, 8)	2.34	(12, 13)	-0.4	(12, 14)	-0.34	(12, 14)	-0.71
(3, 9)	2.23	(12, 12)	-0.49	(12, 12)	-0.38	(12, 12)	-0.73
(3, 4)	1.49	(12, 15)	-0.49	(12, 15)	-0.38	(12, 15)	-0.73
OLMo-2 1B DPO							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(3, 13)	2.77	(13, 14)	0.14	(13, 13)	0.18	(13, 14)	0.12
(3, 14)	2.6	(13, 13)	0.13	(13, 14)	0.18	(13, 13)	0.1
(3, 10)	2.44	(13, 15)	0.13	(13, 15)	0.18	(13, 15)	0.1
(3, 3)	2.43	(14, 14)	0.07	(14, 14)	0.12	(15, 15)	0.0
(3, 15)	2.43	(14, 15)	0.07	(14, 15)	0.12	(14, 14)	-0.01
(3, 11)	2.41	(15, 15)	0.0	(15, 15)	0.0	(14, 15)	-0.01
(3, 12)	2.41	(12, 14)	-0.58	(12, 13)	-0.35	(12, 14)	-0.4
(3, 8)	2.07	(12, 13)	-0.64	(12, 14)	-0.35	(12, 12)	-0.42
(3, 9)	1.9	(12, 12)	-0.67	(12, 12)	-0.4	(12, 15)	-0.42
(5, 6)	1.46	(12, 15)	-0.67	(12, 15)	-0.4	(12, 13)	-0.43
OLMo-2 1B							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(0, 0)	5.49	(1, 14)	1.05	(2, 5)	8.72	(12, 13)	2.5
(0, 15)	5.49	(0, 1)	1.02	(3, 4)	8.69	(12, 14)	2.47
(0, 14)	5.34	(1, 1)	0.98	(3, 5)	8.62	(12, 12)	2.46
(0, 13)	5.22	(1, 15)	0.98	(2, 3)	8.59	(12, 15)	2.46
(1, 2)	5.16	(0, 14)	0.97	(3, 6)	8.41	(1, 7)	2.31
(0, 12)	5.03	(0, 0)	0.93	(2, 7)	8.4	(2, 7)	2.15
(0, 11)	4.97	(0, 15)	0.93	(3, 8)	8.35	(0, 3)	1.85
(1, 1)	4.96	(1, 11)	0.92	(2, 4)	8.29	(0, 7)	1.85
(1, 15)	4.96	(13, 14)	0.89	(3, 7)	8.27	(2, 4)	1.81
(1, 13)	4.86	(1, 9)	0.87	(3, 3)	8.16	(4, 7)	1.8

Table 10: Numerical logit contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

BigBench Tasks - Logit 7B							
OLMo-2 7B SFT							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(15, 24)	2.6	(12, 26)	1.27	(1, 25)	0.79	(15, 16)	3.74
(15, 22)	2.59	(12, 14)	1.22	(1, 24)	0.66	(15, 28)	3.48
(15, 27)	2.56	(12, 27)	1.19	(0, 2)	0.61	(15, 30)	3.48
(15, 23)	2.55	(12, 28)	1.18	(14, 27)	0.61	(15, 15)	3.44
(15, 26)	2.54	(12, 29)	1.15	(22, 27)	0.61	(15, 31)	3.44
(15, 28)	2.51	(12, 30)	1.08	(1, 27)	0.6	(15, 29)	3.42
(15, 21)	2.47	(12, 25)	1.05	(1, 23)	0.59	(15, 18)	3.4
(15, 29)	2.45	(12, 12)	1.0	(1, 28)	0.59	(15, 26)	3.4
(15, 25)	2.43	(12, 24)	1.0	(22, 26)	0.59	(15, 19)	3.37
(15, 30)	2.4	(12, 31)	1.0	(14, 21)	0.58	(15, 27)	3.37
OLMo-2 7B DPO							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(14, 15)	4.11	(12, 26)	2.17	(14, 27)	1.72	(14, 30)	4.47
(14, 27)	3.78	(12, 14)	2.16	(14, 28)	1.67	(14, 14)	4.44
(14, 26)	3.75	(11, 14)	2.15	(14, 29)	1.52	(14, 31)	4.44
(9, 22)	3.72	(12, 27)	2.07	(14, 25)	1.51	(14, 28)	4.43
(14, 22)	3.71	(12, 29)	2.0	(14, 26)	1.5	(14, 29)	4.4
(14, 24)	3.7	(12, 30)	2.0	(14, 21)	1.46	(15, 16)	4.36
(14, 28)	3.7	(12, 24)	1.95	(12, 28)	1.39	(14, 27)	4.34
(14, 23)	3.69	(12, 28)	1.95	(12, 27)	1.31	(14, 19)	4.31
(9, 27)	3.62	(12, 25)	1.93	(14, 22)	1.28	(14, 26)	4.3
(9, 14)	3.6	(12, 21)	1.92	(12, 29)	1.19	(15, 17)	4.3
OLMo-2 7B							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(0, 2)	0.55	(1, 2)	0.5	(14, 27)	1.72	(2, 13)	1.18
(0, 29)	0.38	(1, 1)	0.48	(14, 28)	1.67	(2, 4)	1.07
(0, 10)	0.37	(1, 31)	0.48	(14, 29)	1.52	(2, 25)	1.06
(0, 6)	0.36	(1, 30)	0.44	(14, 25)	1.51	(2, 14)	1.05
(0, 7)	0.34	(1, 27)	0.4	(14, 26)	1.5	(2, 12)	1.04
(0, 8)	0.34	(1, 25)	0.39	(14, 21)	1.46	(2, 23)	1.04
(3, 13)	0.34	(2, 25)	0.38	(12, 28)	1.39	(2, 24)	1.04
(0, 5)	0.32	(2, 2)	0.36	(12, 27)	1.31	(2, 26)	1.01
(0, 30)	0.31	(2, 31)	0.36	(14, 22)	1.28	(2, 15)	1.0
(0, 1)	0.3	(1, 17)	0.34	(12, 29)	1.19	(2, 8)	0.98

Table 11: Numerical logit contributions of OLMo-2 7B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

BigBench Tasks - Rank 7B							
OLMo-2 7B SFT							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(6, 28)	0.04	(21, 24)	0.01	(0, 0)	0.0	(3, 11)	0.14
(6, 29)	0.04	(21, 25)	0.01	(0, 1)	0.0	(8, 10)	0.12
(6, 6)	0.03	(0, 0)	0.0	(0, 2)	0.0	(10, 24)	0.11
(6, 9)	0.03	(0, 1)	0.0	(0, 3)	0.0	(10, 25)	0.11
(6, 23)	0.03	(0, 2)	0.0	(0, 4)	0.0	(10, 10)	0.1
(6, 27)	0.03	(0, 3)	0.0	(0, 5)	0.0	(10, 26)	0.1
(6, 30)	0.03	(0, 4)	0.0	(0, 6)	0.0	(10, 27)	0.1
(6, 31)	0.03	(0, 5)	0.0	(0, 7)	0.0	(10, 28)	0.1
(13, 14)	0.03	(0, 6)	0.0	(0, 8)	0.0	(10, 29)	0.1
(21, 21)	0.03	(0, 7)	0.0	(0, 9)	0.0	(10, 30)	0.1
OLMo-2 7B DPO							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(13, 15)	0.03	(21, 24)	0.01	(0, 12)	0.01	(15, 19)	0.07
(1, 16)	0.02	(0, 0)	0.0	(2, 3)	0.01	(16, 17)	0.07
(6, 9)	0.02	(0, 1)	0.0	(2, 5)	0.01	(16, 18)	0.07
(8, 16)	0.02	(0, 2)	0.0	(2, 6)	0.01	(17, 17)	0.07
(13, 14)	0.02	(0, 3)	0.0	(2, 8)	0.01	(17, 29)	0.07
(0, 16)	0.01	(0, 4)	0.0	(2, 12)	0.01	(17, 30)	0.07
(1, 8)	0.01	(0, 5)	0.0	(2, 13)	0.01	(17, 31)	0.07
(1, 9)	0.01	(0, 6)	0.0	(5, 8)	0.01	(15, 20)	0.06
(1, 10)	0.01	(0, 7)	0.0	(0, 0)	0.0	(15, 15)	0.05
(1, 11)	0.01	(0, 8)	0.0	(0, 1)	0.0	(15, 18)	0.05
OLMo-2 7B							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(6, 11)	0.05	(3, 6)	0.01	(0, 12)	0.01	(7, 19)	0.09
(6, 9)	0.04	(13, 16)	0.01	(2, 3)	0.01	(6, 29)	0.08
(1, 15)	0.03	(0, 0)	0.0	(2, 5)	0.01	(7, 18)	0.08
(1, 16)	0.03	(0, 1)	0.0	(2, 6)	0.01	(7, 20)	0.08
(1, 17)	0.03	(0, 2)	0.0	(2, 8)	0.01	(7, 22)	0.08
(1, 18)	0.03	(0, 3)	0.0	(2, 12)	0.01	(16, 17)	0.08
(2, 3)	0.03	(0, 4)	0.0	(2, 13)	0.01	(6, 6)	0.07
(5, 11)	0.03	(0, 5)	0.0	(5, 8)	0.01	(6, 17)	0.07
(9, 22)	0.03	(0, 6)	0.0	(0, 0)	0.0	(6, 21)	0.07
(1, 1)	0.02	(0, 7)	0.0	(0, 1)	0.0	(6, 30)	0.07

Table 12: Numerical rank contributions of OLMo-2 7B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

BigBench Tasks - Rank 1B							
OLMo-2 1B SFT							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(0, 0)	0.0	(0, 0)	0.0	(0, 0)	0.0	(13, 13)	0.0
(0, 1)	0.0	(0, 1)	0.0	(0, 1)	0.0	(13, 14)	0.0
(0, 2)	0.0	(0, 2)	0.0	(0, 2)	0.0	(13, 15)	0.0
(0, 3)	0.0	(0, 3)	0.0	(0, 3)	0.0	(15, 15)	0.0
(0, 4)	0.0	(0, 4)	0.0	(0, 4)	0.0	(14, 14)	-0.01
(0, 5)	0.0	(0, 5)	0.0	(0, 5)	0.0	(14, 15)	-0.01
(0, 6)	0.0	(0, 6)	0.0	(0, 6)	0.0	(9, 9)	-0.02
(0, 7)	0.0	(0, 7)	0.0	(0, 7)	0.0	(9, 13)	-0.02
(0, 8)	0.0	(0, 8)	0.0	(0, 8)	0.0	(9, 15)	-0.02
(0, 9)	0.0	(0, 9)	0.0	(0, 9)	0.0	(9, 10)	-0.03
OLMo-2 1B DPO							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(1, 3)	0.01	(0, 0)	0.0	(0, 0)	0.0	(10, 10)	0.01
(0, 0)	0.0	(0, 1)	0.0	(0, 1)	0.0	(10, 11)	0.01
(0, 1)	0.0	(0, 2)	0.0	(0, 2)	0.0	(10, 12)	0.01
(0, 2)	0.0	(0, 3)	0.0	(0, 3)	0.0	(10, 13)	0.01
(0, 3)	0.0	(0, 4)	0.0	(0, 4)	0.0	(10, 14)	0.01
(0, 4)	0.0	(0, 5)	0.0	(0, 5)	0.0	(10, 15)	0.01
(0, 5)	0.0	(0, 6)	0.0	(0, 6)	0.0	(9, 9)	0.0
(0, 6)	0.0	(0, 7)	0.0	(0, 7)	0.0	(9, 10)	0.0
(0, 7)	0.0	(0, 8)	0.0	(0, 8)	0.0	(9, 11)	0.0
(0, 8)	0.0	(0, 9)	0.0	(0, 9)	0.0	(9, 12)	0.0
OLMo-2 1B							
metaphor_boolean		implicatures		object_counting		snarks	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(4, 4)	0.13	(0, 0)	0.0	(0, 0)	0.0	(10, 14)	0.1
(4, 5)	0.13	(0, 1)	0.0	(0, 1)	0.0	(3, 6)	0.08
(4, 11)	0.13	(0, 2)	0.0	(0, 2)	0.0	(10, 10)	0.08
(4, 14)	0.13	(0, 3)	0.0	(0, 3)	0.0	(10, 15)	0.08
(4, 15)	0.13	(0, 4)	0.0	(0, 4)	0.0	(2, 9)	0.07
(4, 8)	0.12	(0, 5)	0.0	(0, 5)	0.0	(2, 14)	0.07
(4, 12)	0.12	(0, 6)	0.0	(0, 6)	0.0	(10, 11)	0.07
(4, 13)	0.12	(0, 7)	0.0	(0, 7)	0.0	(10, 12)	0.07
(4, 10)	0.11	(0, 8)	0.0	(0, 8)	0.0	(10, 13)	0.07
(4, 9)	0.1	(0, 9)	0.0	(0, 9)	0.0	(2, 2)	0.06

Table 13: Numerical rank contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

Contrastive Tasks - Rank 7B							
OLMo-2 7B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(7, 18)	0.08	(6, 8)	0.13	(6, 8)	0.01	(0, 13)	0.06
(8, 15)	0.08	(7, 8)	0.13	(9, 13)	0.01	(0, 4)	0.03
(8, 16)	0.08	(5, 9)	0.12	(9, 14)	0.01	(0, 5)	0.03
(8, 17)	0.08	(6, 9)	0.12	(2, 16)	-0.0	(0, 6)	0.03
(9, 24)	0.08	(7, 9)	0.12	(2, 17)	-0.0	(0, 0)	0.02
(4, 14)	0.07	(8, 8)	0.12	(3, 15)	-0.0	(0, 7)	0.02
(4, 19)	0.07	(8, 27)	0.12	(4, 13)	-0.0	(0, 12)	0.02
(7, 17)	0.07	(8, 28)	0.12	(4, 14)	-0.0	(0, 31)	0.02
(8, 18)	0.07	(8, 30)	0.12	(5, 16)	-0.0	(1, 13)	0.02
(4, 15)	0.06	(8, 31)	0.12	(6, 6)	-0.0	(0, 1)	0.01
OLMo-2 7B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(8, 26)	0.1	(6, 9)	0.12	(8, 14)	0.02	(0, 13)	0.04
(8, 25)	0.09	(7, 9)	0.12	(9, 13)	0.02	(0, 5)	0.03
(1, 17)	0.07	(5, 9)	0.11	(9, 14)	0.02	(0, 0)	0.02
(7, 27)	0.07	(8, 8)	0.11	(5, 18)	0.01	(0, 4)	0.02
(7, 28)	0.07	(8, 31)	0.11	(6, 8)	0.01	(0, 12)	0.02
(8, 15)	0.07	(4, 19)	0.1	(6, 14)	0.01	(0, 26)	0.02
(8, 16)	0.07	(7, 8)	0.1	(6, 16)	0.01	(0, 28)	0.02
(8, 17)	0.07	(8, 9)	0.1	(6, 18)	0.01	(0, 31)	0.02
(8, 18)	0.07	(9, 9)	0.1	(9, 11)	0.01	(0, 1)	0.01
(8, 24)	0.07	(9, 28)	0.1	(9, 12)	0.01	(0, 3)	0.01
OLMo-2 7B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(7, 18)	0.1	(3, 18)	0.11	(9, 12)	0.01	(0, 11)	0.06
(2, 24)	0.08	(7, 10)	0.11	(9, 13)	0.01	(0, 10)	0.03
(2, 25)	0.08	(3, 12)	0.1	(9, 14)	0.01	(0, 12)	0.03
(4, 18)	0.08	(4, 12)	0.1	(2, 16)	0.0	(0, 6)	0.02
(7, 17)	0.08	(4, 18)	0.1	(4, 5)	0.0	(0, 0)	0.01
(1, 3)	0.07	(5, 11)	0.1	(5, 5)	0.0	(0, 5)	0.01
(2, 16)	0.07	(5, 17)	0.1	(5, 12)	0.0	(0, 7)	0.01
(2, 27)	0.07	(5, 18)	0.1	(5, 28)	0.0	(0, 9)	0.01
(7, 25)	0.07	(5, 27)	0.1	(5, 31)	0.0	(0, 13)	0.01
(7, 29)	0.07	(6, 18)	0.1	(6, 6)	0.0	(0, 14)	0.01

Table 14: Numerical rank contributions of OLMo-2 7B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

Contrastive Tasks - Logit 1B							
OLMo-2 1B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(3, 15)	1.73	(5, 11)	1.46	(3, 11)	3.15	(0, 11)	4.18
(3, 14)	1.47	(4, 11)	1.42	(5, 13)	3.09	(3, 11)	3.95
(7, 15)	1.44	(7, 11)	1.39	(5, 12)	3.02	(0, 14)	3.8
(3, 13)	1.39	(3, 13)	1.37	(7, 13)	2.99	(0, 13)	3.7
(5, 15)	1.29	(6, 11)	1.37	(6, 13)	2.93	(0, 15)	3.45
(8, 15)	1.13	(5, 15)	1.36	(6, 12)	2.86	(0, 12)	3.34
(6, 15)	1.08	(3, 11)	1.35	(7, 12)	2.85	(2, 11)	3.32
(7, 13)	1.02	(8, 11)	1.32	(3, 13)	2.84	(4, 11)	3.3
(7, 14)	0.95	(9, 11)	1.32	(4, 11)	2.66	(0, 10)	3.22
(9, 15)	0.94	(3, 15)	1.29	(3, 12)	2.63	(5, 11)	3.13
OLMo-2 1B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(3, 15)	2.24	(3, 13)	2.11	(5, 13)	3.04	(0, 11)	3.73
(3, 13)	1.84	(3, 12)	2.09	(7, 13)	2.99	(3, 11)	3.44
(0, 15)	1.79	(4, 11)	1.99	(6, 13)	2.93	(0, 13)	3.39
(3, 14)	1.79	(5, 11)	1.99	(5, 12)	2.77	(0, 14)	3.31
(7, 15)	1.7	(0, 11)	1.98	(3, 11)	2.73	(0, 15)	3.08
(5, 15)	1.49	(9, 11)	1.95	(6, 12)	2.64	(0, 12)	2.97
(8, 15)	1.44	(6, 11)	1.94	(7, 12)	2.59	(5, 11)	2.84
(9, 15)	1.3	(3, 11)	1.93	(8, 13)	2.53	(4, 11)	2.83
(4, 15)	1.23	(2, 11)	1.91	(7, 14)	2.45	(2, 11)	2.8
(6, 15)	1.2	(7, 11)	1.91	(3, 13)	2.38	(6, 11)	2.76
OLMo-2 1B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(3, 3)	-5.42	(3, 5)	-5.94	(6, 14)	1.15	(13, 14)	-0.4
(1, 3)	-5.6	(6, 15)	-5.96	(6, 13)	1.08	(13, 15)	-0.4
(0, 3)	-6.0	(1, 5)	-6.02	(6, 15)	1.08	(11, 15)	-0.59
(2, 3)	-6.15	(5, 5)	-6.05	(6, 12)	0.75	(12, 15)	-0.64
(5, 5)	-6.17	(2, 5)	-6.22	(6, 7)	0.67	(12, 14)	-0.68
(1, 5)	-6.26	(9, 15)	-6.22	(6, 6)	0.55	(6, 9)	-0.71
(3, 5)	-6.29	(10, 15)	-6.22	(6, 11)	0.5	(13, 13)	-0.71
(4, 5)	-6.33	(11, 15)	-6.24	(6, 10)	0.46	(15, 15)	-0.74
(11, 15)	-6.46	(4, 5)	-6.26	(7, 15)	0.44	(11, 14)	-0.76
(13, 15)	-6.49	(7, 15)	-6.29	(7, 14)	0.37	(12, 12)	-0.77

Table 15: Numerical logit contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

Contrastive Tasks - Logit 7B							
OLMo-2 7B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(1, 17)	-0.84	(4, 16)	-1.75	(0, 31)	0.75	(0, 14)	4.11
(1, 16)	-0.91	(6, 16)	-1.93	(0, 0)	0.62	(1, 14)	3.64
(3, 16)	-1.01	(7, 16)	-1.97	(1, 31)	0.19	(0, 1)	2.66
(3, 17)	-1.07	(8, 16)	-2.0	(0, 1)	0.11	(0, 31)	2.6
(8, 17)	-1.07	(1, 16)	-2.01	(0, 27)	0.05	(1, 15)	2.46
(4, 16)	-1.11	(3, 16)	-2.04	(0, 4)	-0.02	(0, 6)	2.44
(8, 16)	-1.13	(5, 16)	-2.29	(1, 1)	-0.04	(0, 28)	2.35
(0, 17)	-1.16	(9, 16)	-2.29	(0, 28)	-0.1	(0, 13)	2.26
(4, 17)	-1.16	(4, 14)	-2.3	(0, 30)	-0.34	(0, 15)	2.26
(6, 16)	-1.2	(11, 16)	-2.3	(0, 3)	-0.35	(0, 0)	2.25
OLMo-2 7B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(1, 16)	0.87	(4, 16)	-0.12	(0, 0)	1.91	(0, 14)	5.36
(3, 16)	0.81	(1, 16)	-0.27	(0, 28)	1.85	(1, 14)	4.89
(1, 17)	0.7	(3, 16)	-0.32	(0, 31)	1.76	(0, 1)	4.41
(8, 16)	0.64	(8, 16)	-0.33	(0, 27)	1.74	(0, 0)	4.05
(4, 16)	0.56	(7, 16)	-0.35	(0, 29)	1.26	(1, 15)	3.88
(3, 17)	0.46	(6, 16)	-0.44	(0, 1)	1.21	(0, 15)	3.8
(0, 16)	0.45	(0, 16)	-0.58	(0, 26)	1.11	(0, 6)	3.63
(0, 17)	0.45	(11, 16)	-0.67	(1, 31)	1.07	(0, 31)	3.59
(8, 17)	0.4	(9, 16)	-0.7	(0, 30)	0.97	(1, 1)	3.44
(7, 16)	0.36	(4, 14)	-0.75	(0, 9)	0.94	(0, 5)	3.4
OLMo-2 7B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.	Layer(s)	Logit contrib.
(0, 0)	-2.3	(0, 0)	-3.0	(31, 31)	-0.43	(31, 31)	0.66
(1, 3)	-6.21	(7, 19)	-4.92	(2, 31)	-1.97	(0, 6)	0.18
(3, 3)	-6.23	(7, 20)	-4.93	(0, 1)	-2.16	(0, 10)	-0.18
(2, 3)	-6.34	(7, 21)	-4.93	(5, 31)	-2.23	(0, 7)	-0.2
(0, 3)	-6.51	(9, 21)	-5.03	(0, 0)	-2.4	(0, 8)	-0.28
(5, 7)	-6.51	(8, 21)	-5.04	(30, 31)	-2.41	(0, 9)	-0.28
(6, 7)	-6.51	(0, 1)	-5.07	(0, 31)	-2.43	(0, 11)	-0.36
(4, 7)	-6.53	(6, 21)	-5.07	(1, 31)	-2.5	(0, 1)	-0.41
(5, 5)	-6.54	(11, 20)	-5.08	(30, 30)	-2.68	(0, 31)	-0.59
(7, 7)	-6.58	(11, 21)	-5.08	(1, 1)	-2.79	(0, 5)	-0.69

Table 16: Numerical logit contributions of OLMo-2 7B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

Contrastive Tasks - Logits 1B							
OLMo-2 1B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.
(3, 14, 15)	1.97	(2, 5, 11)	1.69	(3, 6, 12)	4.4	(0, 11, 14)	4.42
(3, 13, 15)	1.87	(4, 5, 11)	1.62	(3, 11, 13)	4.36	(0, 6, 11)	4.29
(3, 11, 15)	1.67	(3, 13, 15)	1.61	(3, 6, 13)	4.25	(0, 11, 15)	4.25
(3, 12, 15)	1.63	(2, 4, 11)	1.58	(1, 6, 12)	4.09	(3, 11, 14)	4.22
(7, 13, 15)	1.5	(3, 14, 15)	1.57	(3, 11, 14)	4.09	(0, 7, 11)	4.16
(5, 7, 15)	1.49	(5, 9, 11)	1.53	(3, 11, 12)	4.04	(3, 5, 11)	4.14
(7, 14, 15)	1.45	(5, 10, 11)	1.5	(3, 7, 12)	4.02	(0, 5, 11)	4.06
(5, 8, 15)	1.41	(1, 6, 11)	1.49	(3, 7, 13)	3.98	(0, 10, 11)	4.04
(3, 13, 14)	1.39	(4, 6, 11)	1.48	(3, 10, 13)	3.98	(0, 11, 13)	4.04
(7, 11, 15)	1.36	(7, 9, 11)	1.48	(4, 6, 12)	3.93	(3, 4, 11)	4.02
OLMo-2 1B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.
(3, 14, 15)	2.51	(3, 13, 15)	2.19	(3, 11, 13)	4.56	(3, 5, 11)	3.76
(3, 13, 15)	2.44	(2, 5, 11)	2.11	(3, 6, 13)	4.14	(0, 11, 14)	3.74
(3, 11, 15)	2.22	(2, 4, 11)	2.09	(3, 11, 14)	4.13	(0, 11, 15)	3.73
(3, 12, 15)	2.21	(3, 14, 15)	2.07	(3, 6, 12)	4.07	(0, 6, 11)	3.69
(0, 14, 15)	1.88	(4, 5, 11)	2.06	(2, 11, 13)	3.95	(3, 4, 11)	3.68
(2, 3, 15)	1.85	(5, 9, 11)	2.06	(4, 6, 13)	3.93	(0, 11, 13)	3.64
(7, 13, 15)	1.8	(4, 6, 11)	2.02	(3, 11, 12)	3.92	(2, 4, 11)	3.56
(3, 13, 14)	1.73	(5, 10, 11)	2.01	(3, 10, 13)	3.9	(0, 7, 11)	3.55
(7, 14, 15)	1.7	(7, 9, 11)	2.01	(3, 7, 13)	3.89	(3, 11, 14)	3.48
(0, 13, 15)	1.69	(5, 6, 11)	2.0	(0, 11, 13)	3.86	(0, 5, 11)	3.47
OLMo-2 1B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.	Layer(s)	Logits contrib.
(1, 2, 3)	-5.87	(3, 4, 5)	-5.83	(6, 7, 14)	1.51	(6, 8, 11)	0.13
(0, 1, 3)	-6.0	(7, 9, 15)	-6.02	(6, 7, 13)	1.48	(6, 7, 11)	0.03
(0, 2, 3)	-6.2	(7, 10, 15)	-6.08	(6, 7, 15)	1.45	(6, 8, 10)	0.01
(3, 4, 5)	-6.21	(6, 14, 15)	-6.12	(6, 7, 12)	1.21	(6, 9, 11)	-0.15
(6, 9, 15)	-6.4	(7, 11, 15)	-6.16	(6, 10, 13)	1.2	(6, 9, 15)	-0.27
(0, 1, 5)	-6.41	(1, 2, 5)	-6.18	(6, 10, 14)	1.16	(6, 7, 10)	-0.34
(11, 12, 15)	-6.41	(6, 9, 15)	-6.18	(6, 12, 14)	1.14	(6, 9, 12)	-0.34
(7, 9, 15)	-6.42	(10, 11, 15)	-6.19	(6, 13, 14)	1.14	(6, 9, 10)	-0.35
(1, 4, 5)	-6.45	(9, 11, 15)	-6.2	(6, 13, 15)	1.13	(6, 9, 14)	-0.38
(1, 2, 5)	-6.48	(2, 4, 5)	-6.21	(6, 14, 15)	1.12	(13, 14, 15)	-0.4

Table 17: Numerical logit contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

Contrastive Tasks - Rank 1B							
OLMo-2 1B SFT							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(3, 14, 15)	0.14	(0, 14, 15)	0.07	(3, 5, 13)	0.18	(2, 3, 7)	0.06
(3, 11, 13)	0.11	(2, 3, 14)	0.07	(3, 6, 12)	0.18	(2, 6, 9)	0.05
(3, 11, 15)	0.11	(4, 6, 15)	0.07	(4, 5, 13)	0.17	(3, 6, 7)	0.05
(3, 13, 14)	0.11	(2, 3, 15)	0.06	(3, 5, 12)	0.16	(3, 7, 14)	0.05
(3, 13, 15)	0.11	(3, 14, 15)	0.06	(3, 6, 13)	0.16	(3, 7, 15)	0.05
(0, 14, 15)	0.1	(4, 5, 6)	0.06	(4, 6, 12)	0.16	(0, 3, 7)	0.04
(2, 3, 15)	0.1	(4, 6, 14)	0.06	(3, 11, 13)	0.15	(1, 6, 7)	0.04
(3, 11, 14)	0.1	(2, 3, 13)	0.05	(4, 6, 13)	0.15	(2, 4, 7)	0.04
(2, 3, 14)	0.09	(2, 5, 13)	0.05	(4, 7, 13)	0.15	(2, 4, 9)	0.04
(3, 5, 11)	0.09	(2, 5, 14)	0.05	(2, 6, 12)	0.14	(3, 4, 7)	0.04
OLMo-2 1B DPO							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(3, 14, 15)	0.13	(0, 14, 15)	0.07	(3, 5, 13)	0.17	(3, 4, 6)	0.04
(3, 13, 14)	0.11	(4, 6, 14)	0.04	(3, 6, 13)	0.16	(3, 4, 7)	0.04
(3, 13, 15)	0.11	(2, 3, 14)	0.03	(3, 11, 13)	0.16	(3, 7, 15)	0.04
(0, 14, 15)	0.09	(2, 3, 15)	0.03	(4, 6, 12)	0.15	(0, 5, 7)	0.03
(3, 11, 13)	0.08	(3, 14, 15)	0.03	(4, 6, 13)	0.15	(1, 6, 7)	0.03
(3, 11, 14)	0.08	(4, 5, 6)	0.03	(3, 5, 12)	0.14	(1, 7, 15)	0.03
(3, 11, 15)	0.08	(4, 6, 13)	0.03	(3, 6, 12)	0.14	(2, 3, 7)	0.03
(0, 3, 15)	0.07	(5, 6, 14)	0.03	(4, 5, 13)	0.14	(2, 4, 7)	0.03
(0, 13, 14)	0.07	(5, 6, 15)	0.03	(0, 13, 14)	0.13	(2, 4, 8)	0.03
(0, 13, 15)	0.07	(0, 1, 3)	0.02	(2, 5, 12)	0.13	(3, 4, 8)	0.03
OLMo-2 1B							
adj: ant		adj: comp		anim: color		anim: can_fly	
Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.	Layer(s)	Rank contrib.
(1, 14, 15)	0.16	(1, 14, 15)	0.08	(6, 7, 13)	0.16	(5, 8, 10)	0.22
(0, 1, 14)	0.14	(6, 14, 15)	0.07	(6, 7, 14)	0.16	(4, 8, 10)	0.21
(0, 1, 13)	0.13	(7, 8, 9)	0.07	(6, 13, 14)	0.16	(4, 8, 11)	0.21
(0, 1, 15)	0.13	(7, 9, 14)	0.07	(6, 7, 12)	0.15	(4, 7, 10)	0.19
(6, 7, 9)	0.12	(7, 9, 15)	0.07	(6, 13, 15)	0.15	(5, 7, 10)	0.19
(6, 9, 14)	0.12	(7, 9, 10)	0.06	(6, 7, 8)	0.14	(3, 8, 11)	0.18
(1, 13, 15)	0.11	(7, 9, 11)	0.06	(6, 7, 11)	0.14	(4, 8, 14)	0.18
(7, 9, 14)	0.11	(7, 10, 11)	0.06	(6, 7, 15)	0.14	(5, 8, 14)	0.18
(0, 1, 6)	0.1	(0, 1, 14)	0.05	(6, 14, 15)	0.13	(3, 8, 14)	0.17
(1, 6, 15)	0.1	(1, 13, 14)	0.05	(6, 7, 9)	0.12	(4, 7, 8)	0.17

Table 18: Numerical rank contributions of OLMo-2 1B models for various patching configurations. For each model, the 10 highest-contributing combinations are shown.

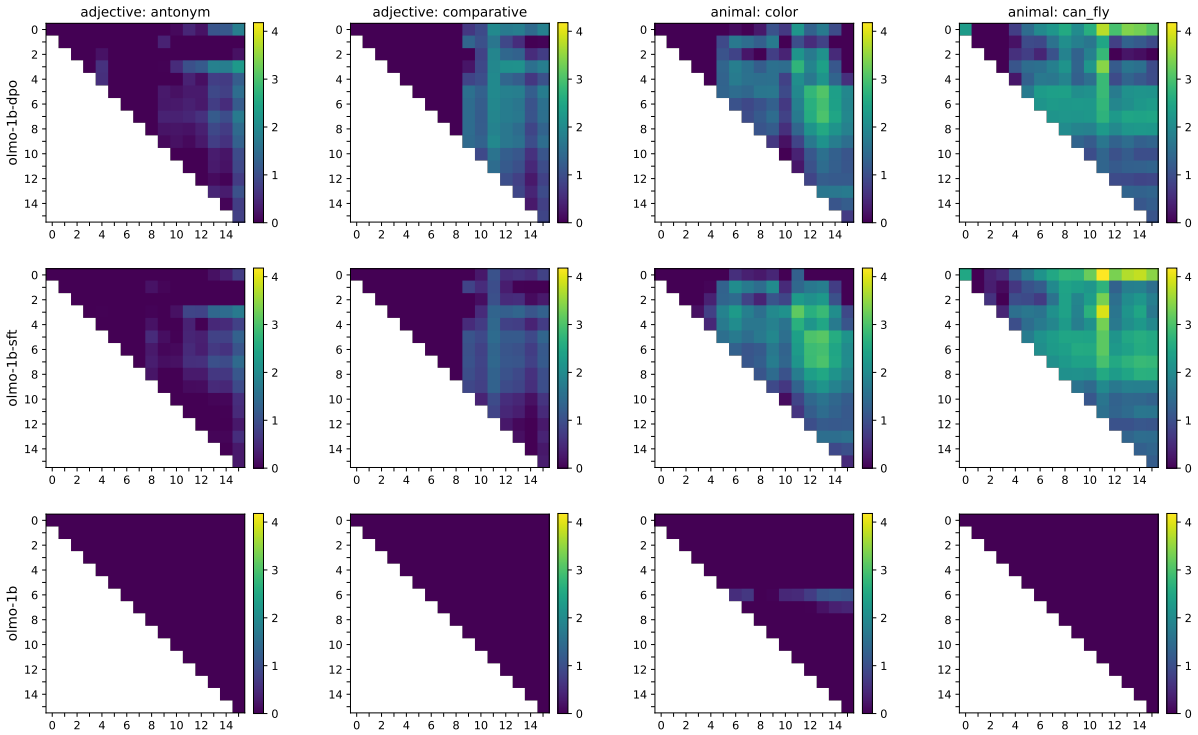

Figure 10: Effects of 1- and 2-layer patching configurations on the logit of the target token. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching. Two important properties are shown. *Localization:* Across all tasks, we observe localized points where the logit improvement is the greatest. *Superadditivity:* Two-layer combinations bring greater improvements than single layers.

stabilized and that few heads change significantly at this point (Table 21).

Tasks. The similarity scores for head activity between the two contrastive ADJECTIVES is shown in Tables 22 - 24. In accordance with our result that increasing k also increases general head activity, we find that the similarity increases with k for most of our selected layer groups. However, across k, we notice that Layers 0-2 have the highest similarity in activity at the T_{inst} token position, and that the lowest similarity scores are observed somewhere in the later layer groups (6-9 or 10-15). Moreover, we observe that this high similarity does not occur before T_{inst}, and seldom occurs after. This indicates that the T_{inst} position is the location of a particular kind of computation that does not occur at the other token positions.

F.3 Initial Generalization to the ANIMALS Tasks

In Figures 41 - 46, we present the attention head activity patterns for ANIMALS: COLOR and ANIMALS: CAN_FLY, with k=1. Due to resource constraints, we do not conduct path analysis on these tasks. However, we observe that our overall conclusions for the ADJECTIVES tasks hold, with some interesting insights regarding eager instruction representations, as discussed in Section 6.1.

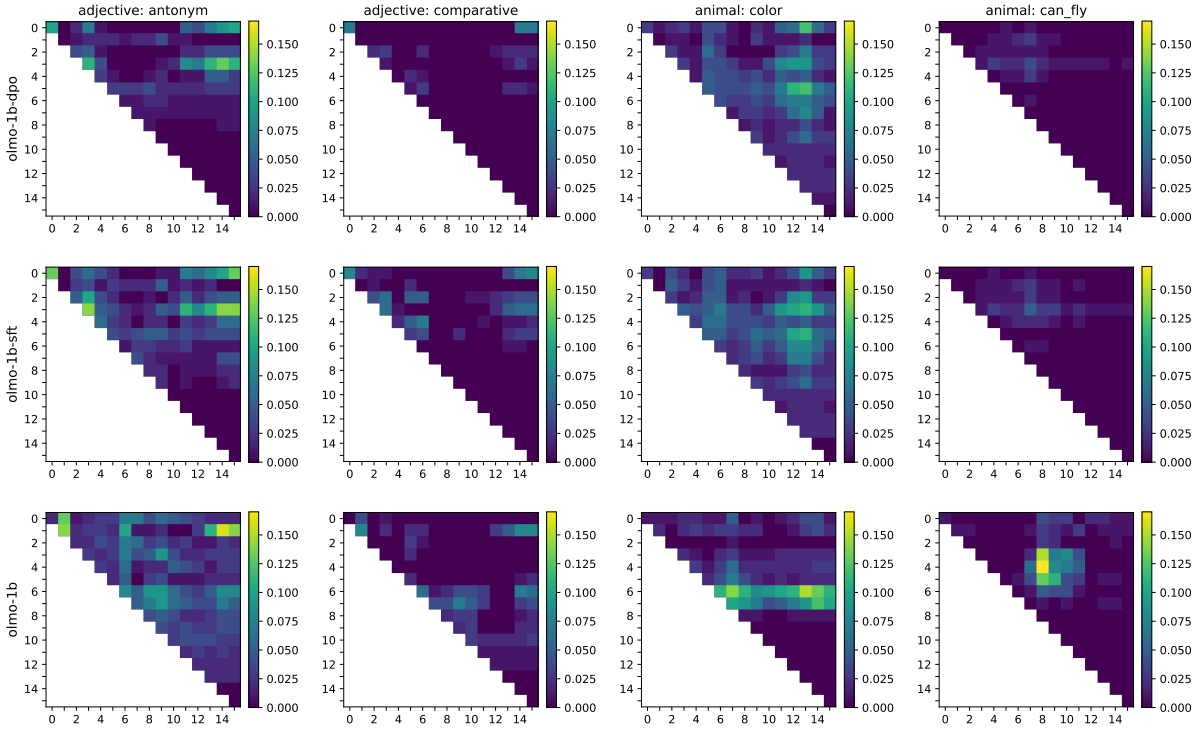

Figure 11: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token for OLMo-2 1B models. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

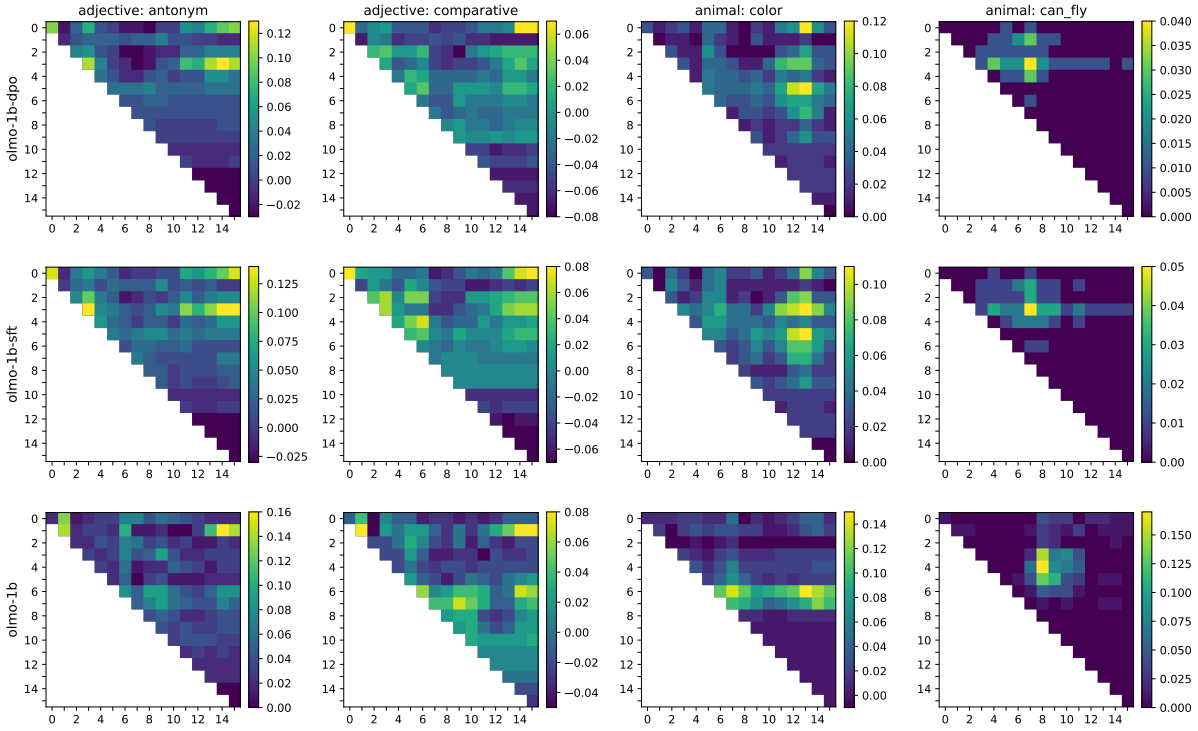

Figure 12: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token for OLMo-2 1B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

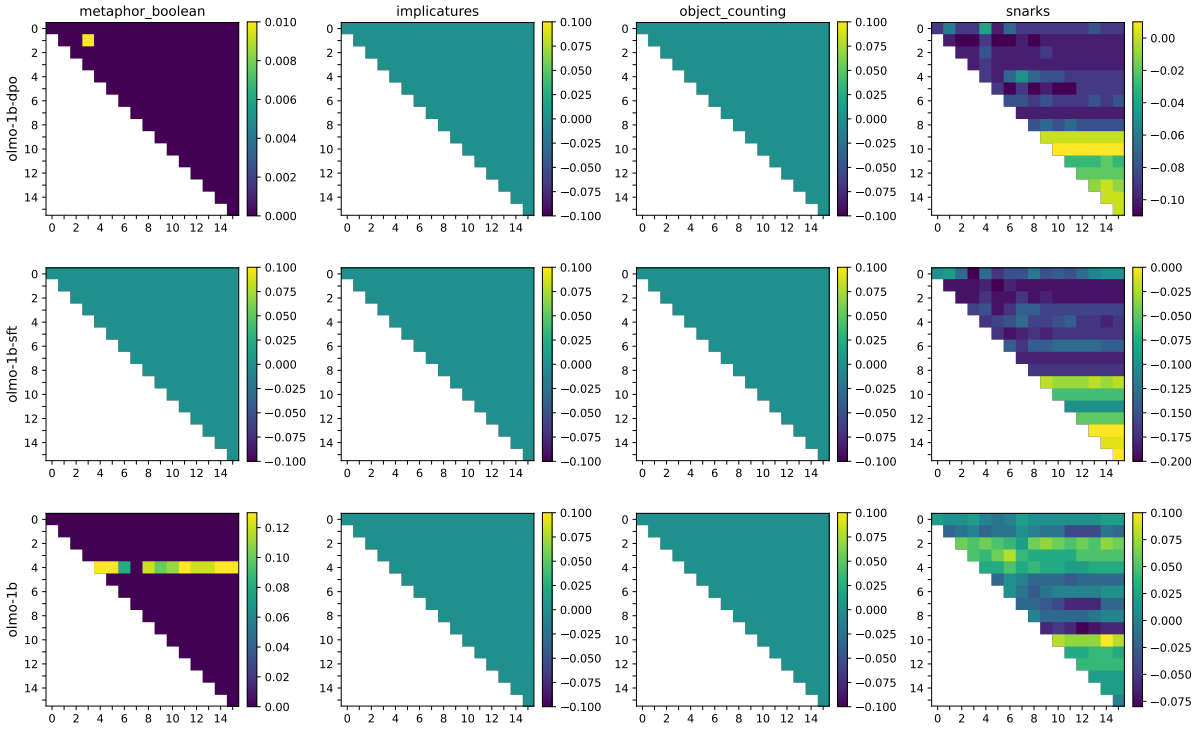

Figure 13: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token for OLMo-2 1B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

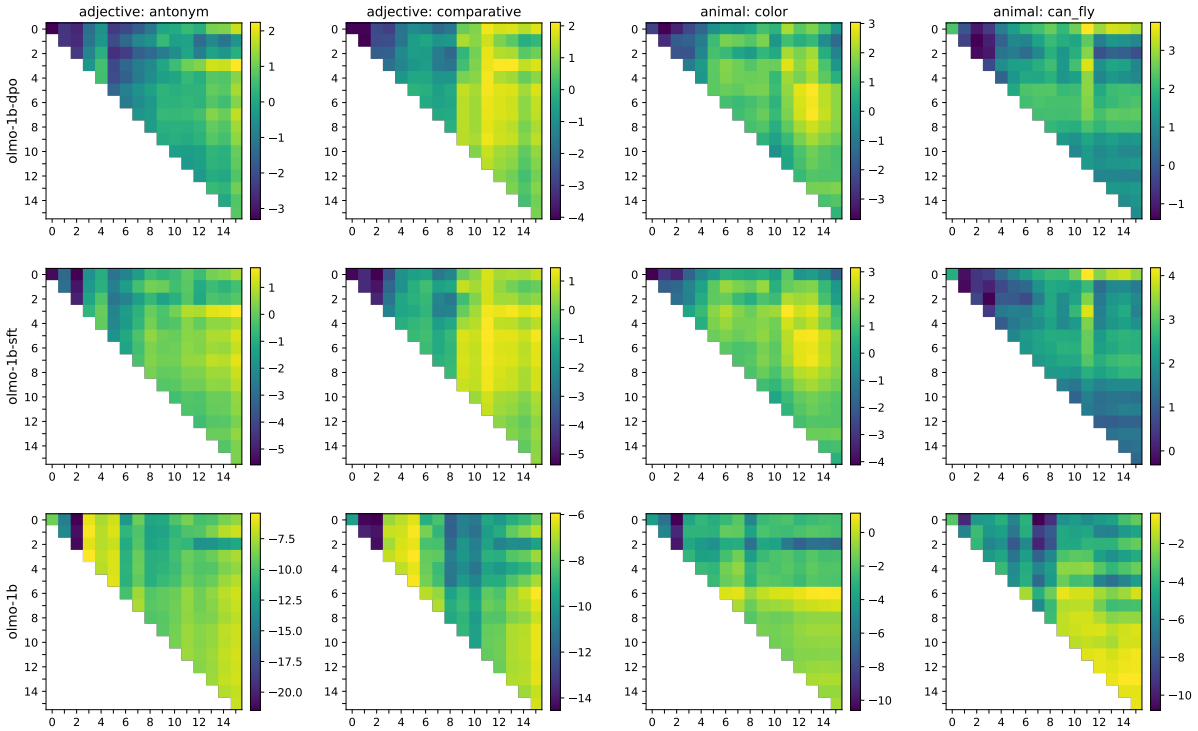

Figure 14: Effects of 1- and 2-layer patching configurations on the logit of the target token for OLMo-2 1B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

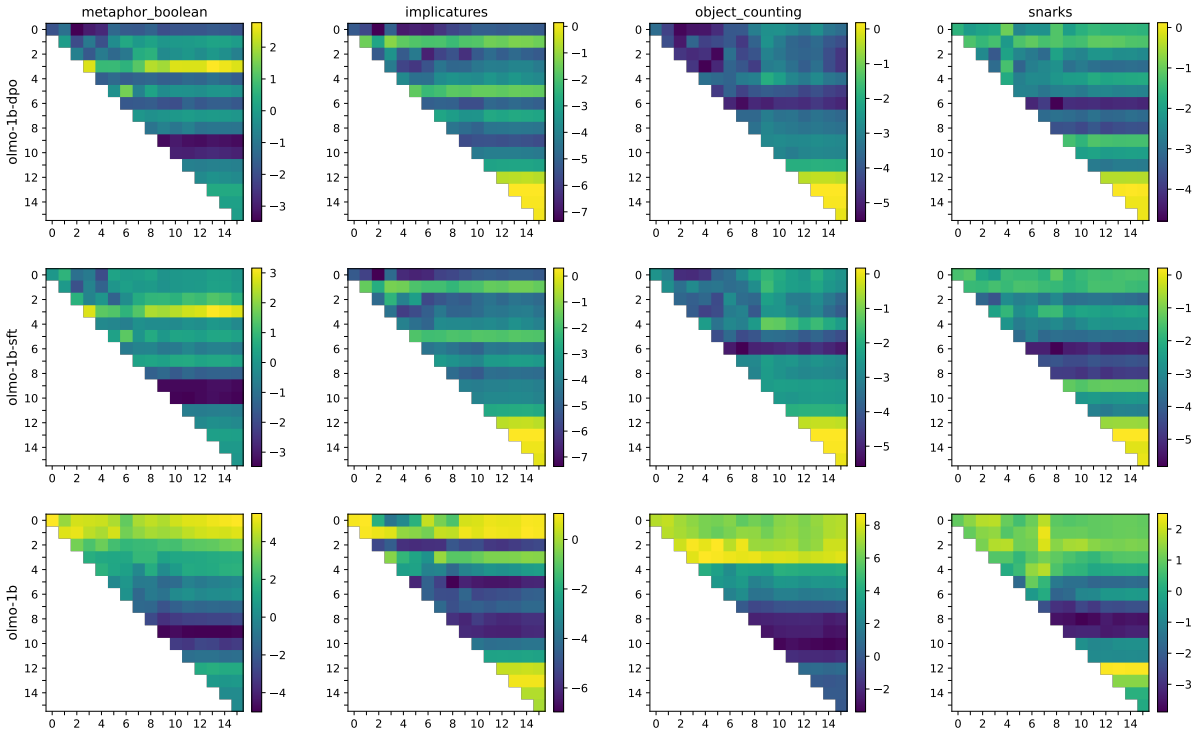

Figure 15: Effects of 1- and 2-layer patching configurations on the logit of the target token for OLMo-2 1B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

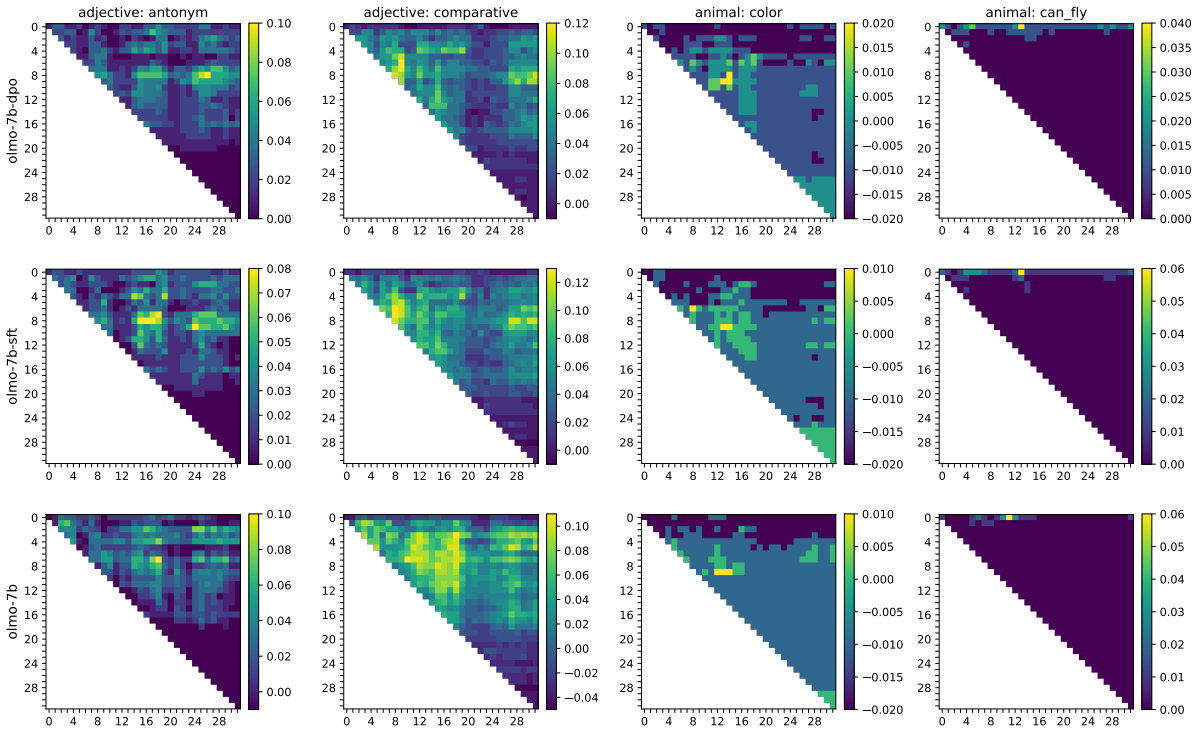

Figure 16: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token for OLMo-2 7B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

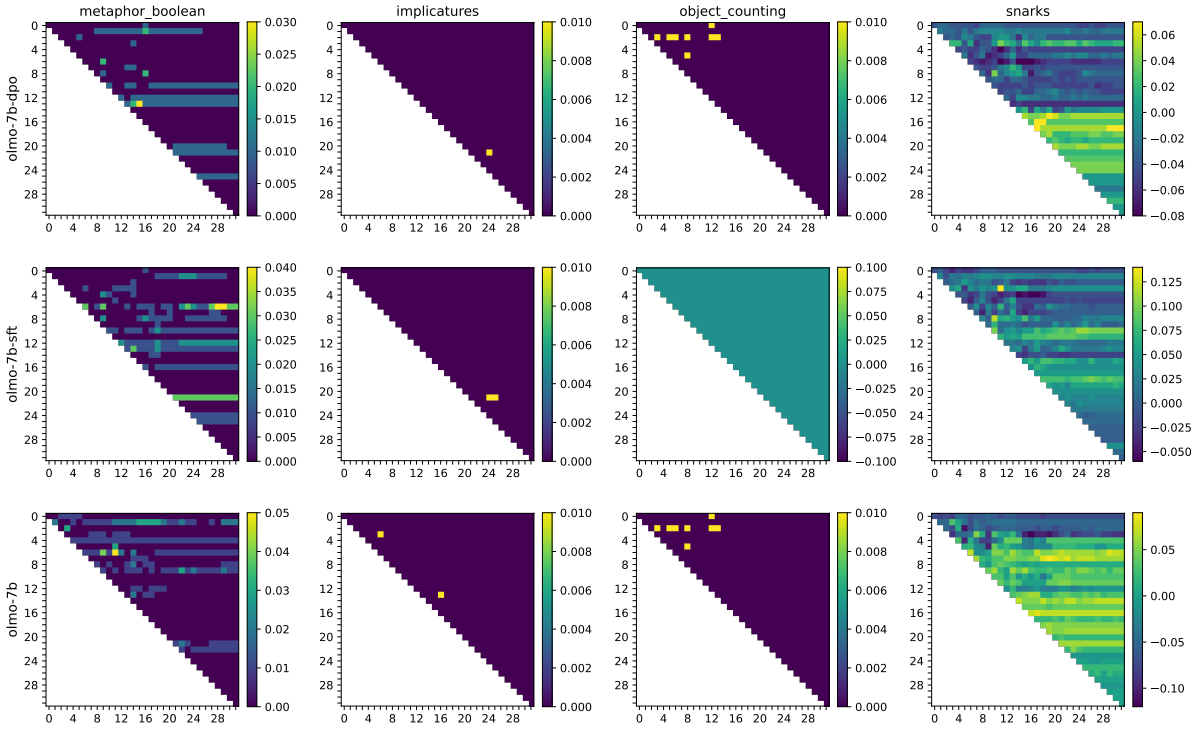

Figure 17: Effects of 1- and 2-layer patching configurations on the reciprocal rank of the target token for OLMo-2 7B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

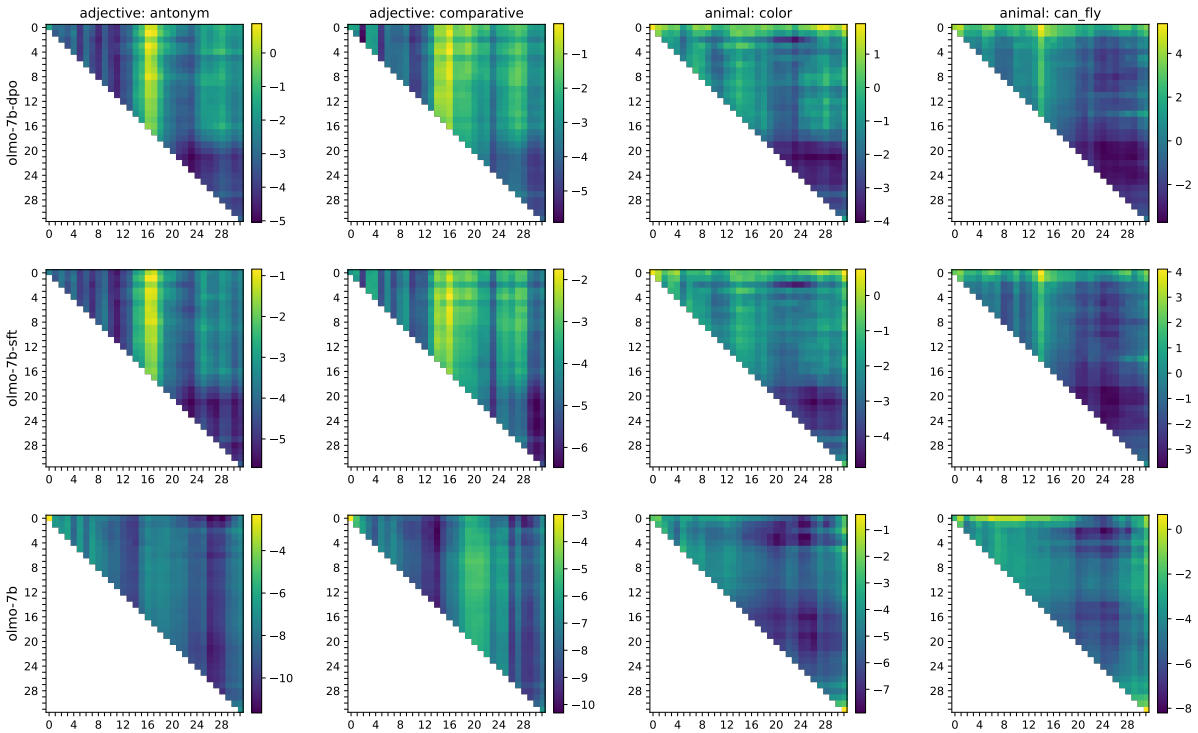

Figure 18: Effects of 1- and 2-layer patching configurations on the logit of the target token for OLMo-2 7B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

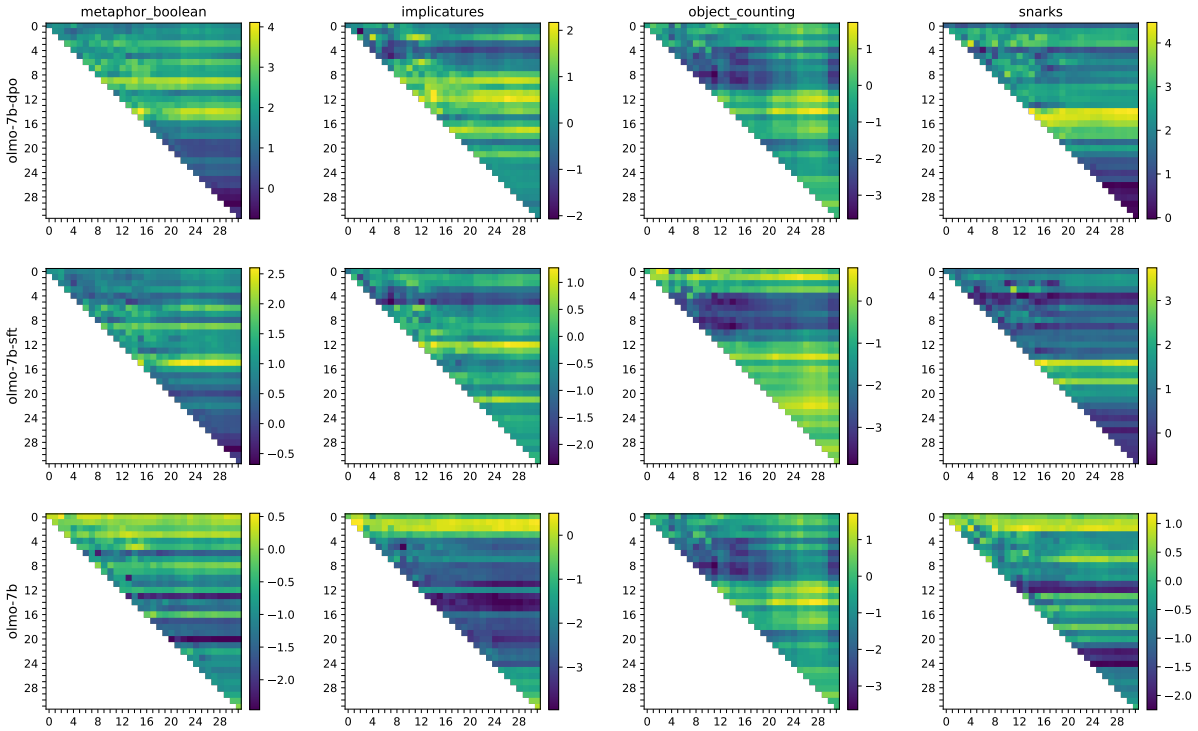

Figure 19: Effects of 1- and 2-layer patching configurations on the logit of the target token for OLMo-2 7B models, without normalization across panels. Each square of the x- and y-coordinate grid represents the corresponding layers of the model being patched. Coordinates where x=y represent 1-layer patching.

Task	Prompt
adjective: comparative	'You will be given a task instruction and an answer someone wrote in response. Regardless of whether the answer is correct, determine whether the answer is **an English adjective in the comparative form.** Output either 'yes' or 'no'. Instruction: '{}' Answer: '{}'. Your decision:'
adjective: antonym	'You will be given a task instruction and an answer someone wrote in response. Regardless of whether the answer is correct, determine whether the answer is **an English adjective in its standard declarative form.** Output either 'yes' or 'no'. Instruction: '{}' Answer: '{}'. Your decision:'
animal: color	'You will be given a task instruction and an answer someone wrote in response. Regardless of whether the answer is correct, determine whether the answer is **the name of a color.** Output either 'yes' or 'no'. Instruction: '{}' Answer: '{}'. Your decision:'

Table 19: LLM judge prompts for instructional accuracy (IA). The 'instruction' field is filled with the query-independent task instruction and the 'answer' field is filled with the inference model's output given a query. Thus, IA measures the appropriateness of a response to the instruction rather than the correctness of the answer.

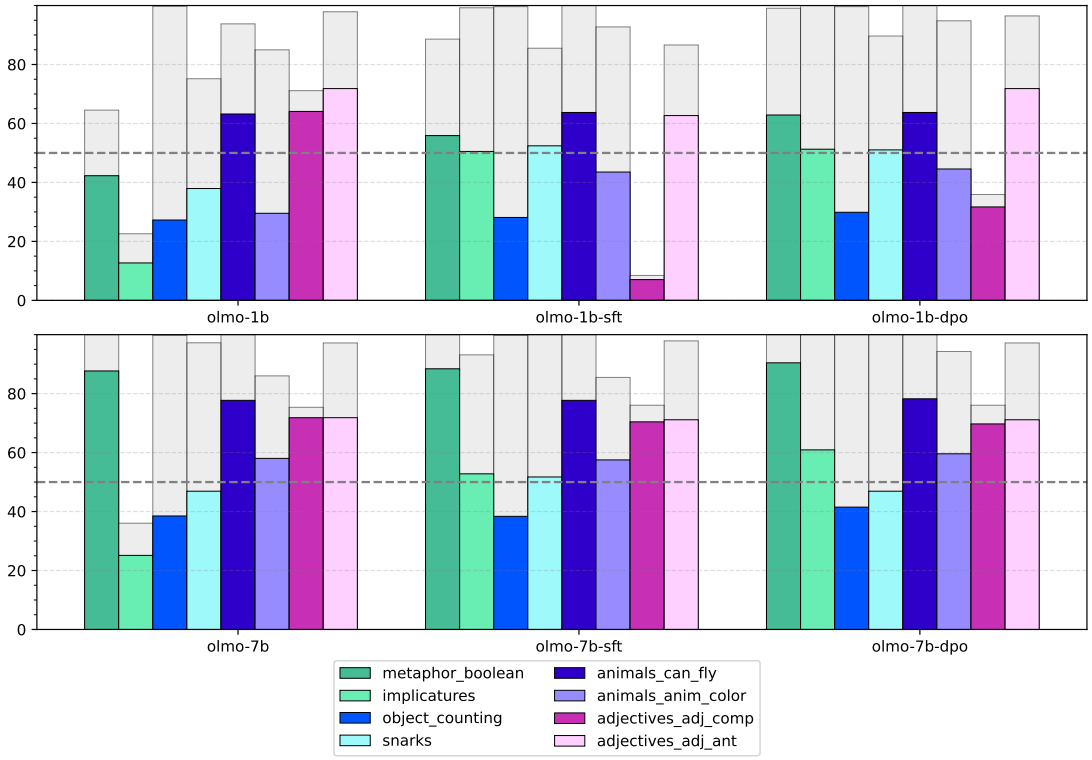

Figure 20: EMA scores (in color) for OLMo-1B and OLMo-7B models. Instruction Accuracy scores (gray) show that instruction-following abilities are present for most models and tasks (IA>50%)

	adj_comp			adj_ant		
Model	L0–1	L4–5	L14–15	L0–1	L4–5	L14–15
OLMo-1B	**0.91**	0.22	0.19	**0.88**	0.16	0.24
OLMo-1B-DPO	**0.84**	0.20	0.35	**0.81**	0.18	0.31

Table 20: Jaccard similarity of active heads (threshold > 0.1) between k=1 and k=3, by layer group. The activity in Layers 0-1 remains highly similar, while middle and later layer pairs are dissimilar, indicating that the increased k changes activity patterns primarily in these layers.

	adj_comp			adj_ant		
Model	L0–1	L4–5	L14–15	L0–1	L4–5	L14–15
OLMo-1B	**1.00**	**0.88**	**0.81**	**1.00**	**0.88**	**0.64**
OLMo-1B-DPO	**1.00**	**0.76**	**0.90**	**1.00**	**0.68**	**0.85**

Table 21: Jaccard similarity of active heads (threshold > 0.1) between k=2 and k=3, by layer group. The high similarity scores indicate that there is little change in head activity from k=2 to k=3.

$T_{\text{inst}} - 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.375	0.000	0.714
OLMo-1B-SFT	0.333	0.000	0.500
OLMo-1B-DPO	0.000	0.000	0.444
T_{inst}			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.762	0.500	0.714
OLMo-1B-SFT	0.952	0.333	0.727
OLMo-1B-DPO	0.909	0.286	0.500
$T_{\text{inst}} + 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.250	0.500	0.714
OLMo-1B-SFT	0.714	0.250	0.600
OLMo-1B-DPO	0.625	0.500	0.500

Table 22: Jaccard similarity of attention head activity (k=1) between the two contrastive ADJECTIVES tasks at various token positions, over 100 samples.

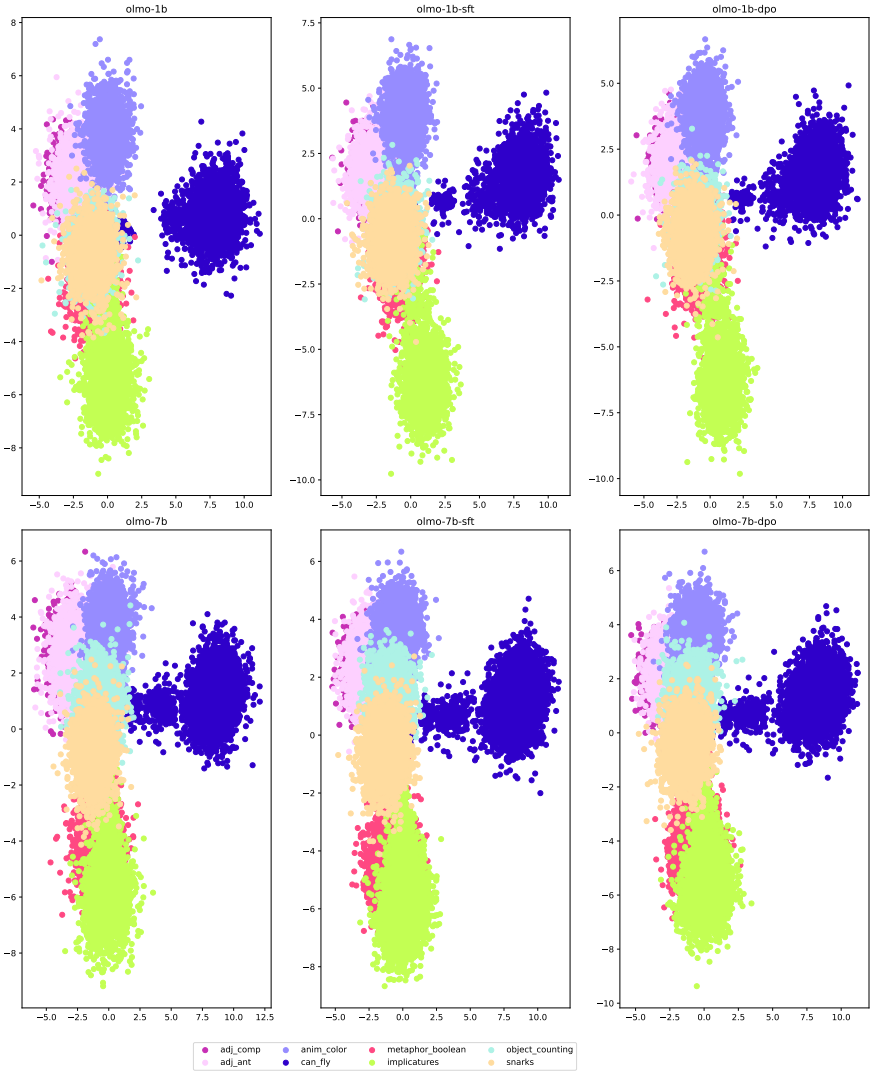

Figure 21: Our LDA analysis of 200 instruction rephrasals on OLMo-2 models shows that tasks form organized clusters, in line with previous results. However, they are not always linearly separable (e.g. "adj_comp" and "adj_ant").

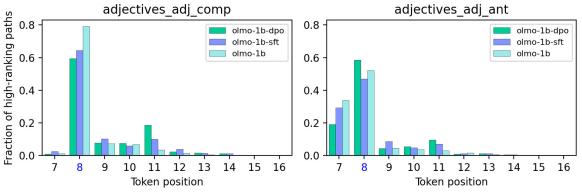

Figure 22: Fraction of high-ranking paths contributed by each token position, across all observed high-ranking paths. Top-k attention was used, with k=1.

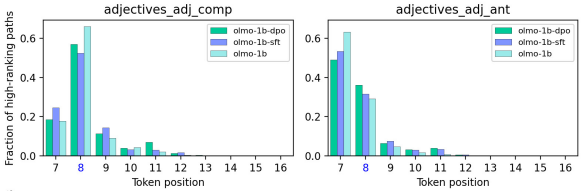

Figure 23: Fraction of high-ranking paths contributed by each token position, across all observed high-ranking paths. Top-k attention was used, with k=2.

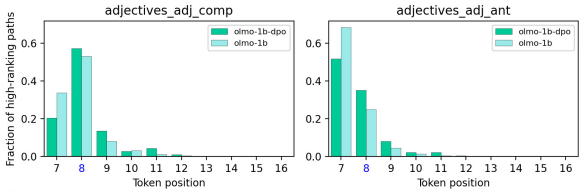

Figure 24: Fraction of high-ranking paths contributed by each token position, across all observed high-ranking paths. Top-k attention was used, with k=3. Due to constraints, the SFT model was not analyzed.

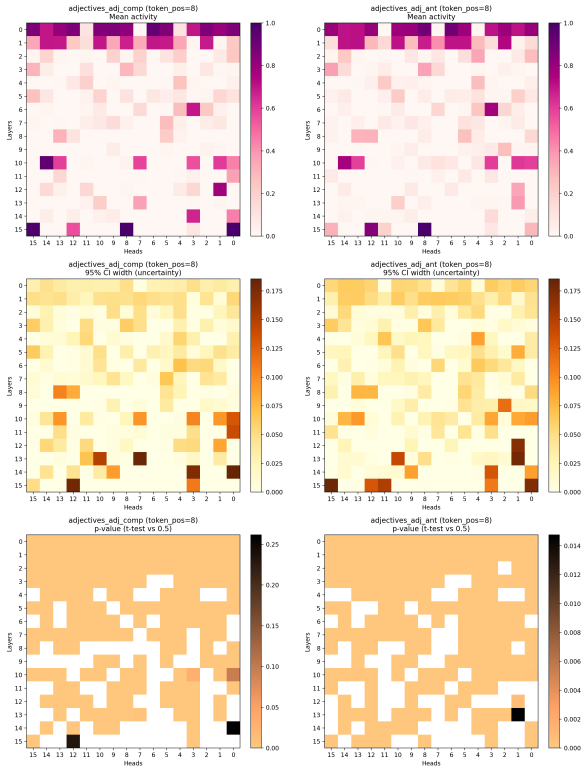

Figure 25: Results of our statistical tests of attention head activity, for OLMo-1B, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

$T_{\text{inst}} - 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.769	0.759	0.577
OLMo-1B-SFT	0.765	0.760	0.792
OLMo-1B-DPO	0.667	0.708	0.826
T_{inst}			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	1.000	0.931	0.577
OLMo-1B-SFT	0.933	0.667	0.750
OLMo-1B-DPO	0.935	0.700	0.826
$T_{\text{inst}} + 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.524	0.583	0.577
OLMo-1B-SFT	0.500	0.308	0.750
OLMo-1B-DPO	0.538	0.429	0.846

Table 23: Jaccard similarity of attention head activity (k=2) between the two contrastive ADJECTIVES tasks at various token positions, over 100 samples.

$T_{\text{inst}} - 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.931	1.000	0.839
OLMo-1B-DPO	0.773	0.800	0.769
T_{inst}			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	1.000	1.000	0.806
OLMo-1B-DPO	0.969	0.679	0.769
$T_{\text{inst}} + 1$			
Model	**Layers 0–2**	**Layers 6–9**	**Layers 10–15**
OLMo-1B	0.577	0.871	0.839
OLMo-1B-DPO	0.684	0.750	0.846

Table 24: Jaccard similarity of attention head activity (k=3) between the two contrastive ADJECTIVES tasks at various token positions, over 100 samples.

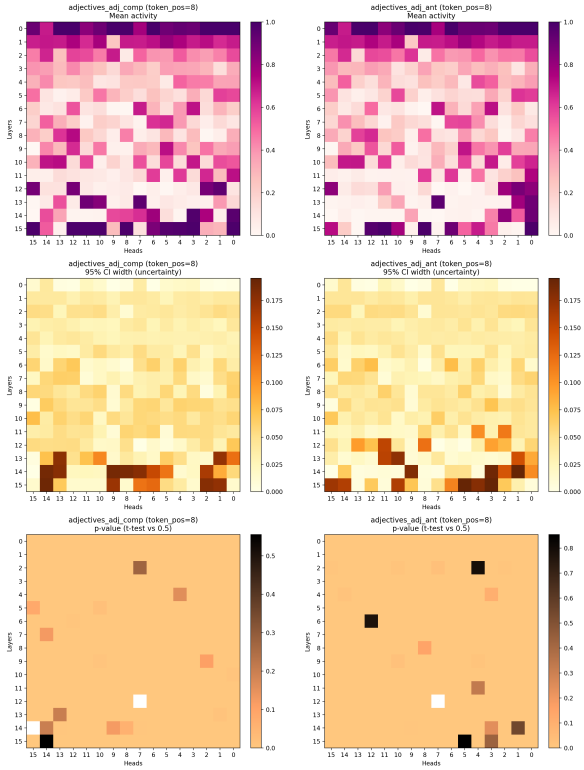

Figure 26: Results of our statistical tests of attention head activity, for OLMo-1B, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

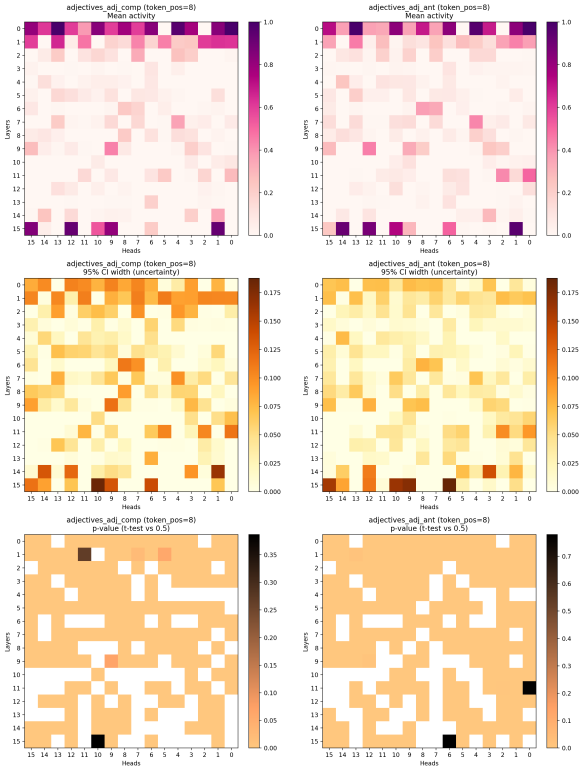

Figure 27: Results of our statistical tests of attention head activity, for OLMo-1B-SFT, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

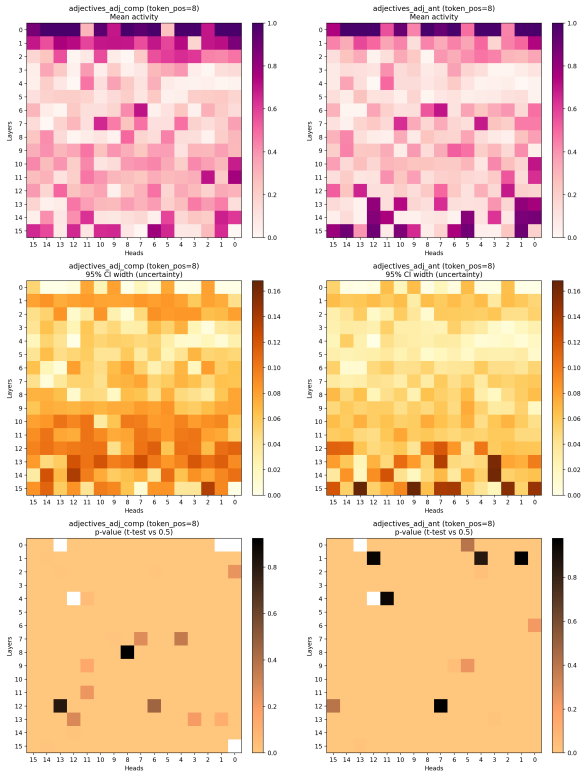

Figure 28: Results of our statistical tests of attention head activity, for OLMo-1B-SFT, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

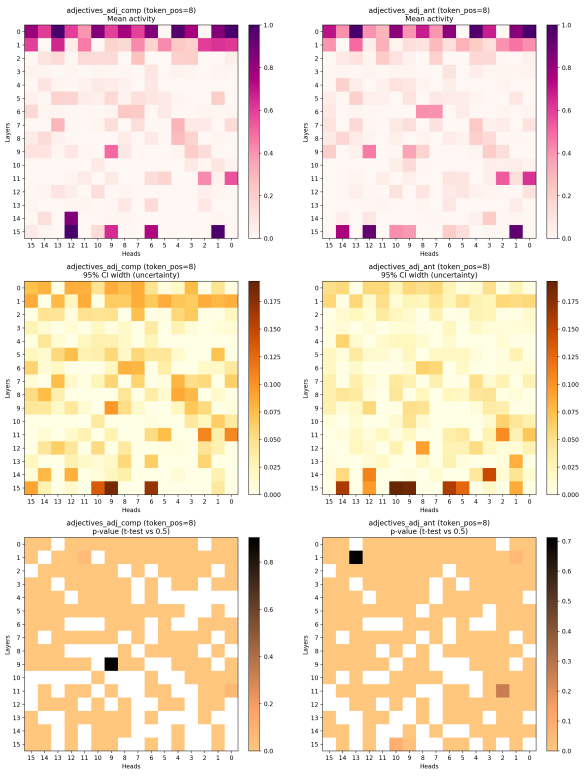

Figure 29: Results of our statistical tests of attention head activity, for OLMo-1B-DPO, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

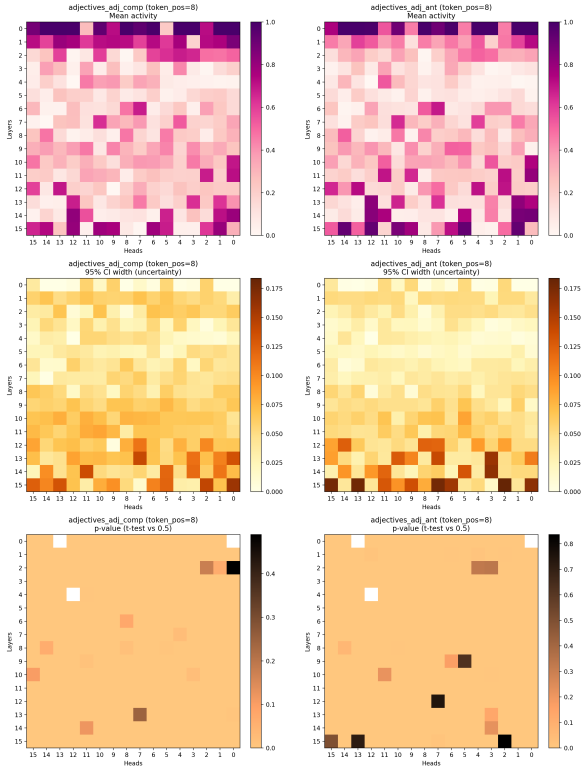

Figure 30: Results of our statistical tests of attention head activity, for OLMo-1B-DPO, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

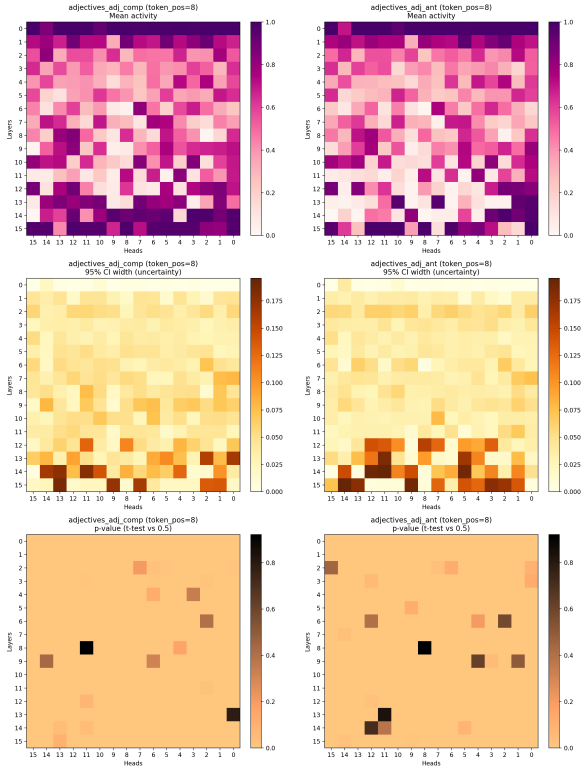

Figure 31: Results of our statistical tests of attention head activity, for OLMo-1B, k=3. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

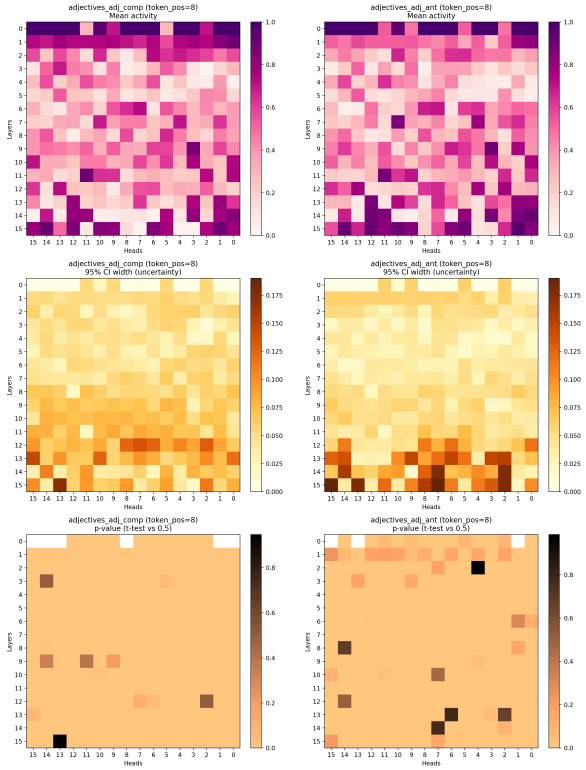

Figure 32: Results of our statistical tests of attention head activity, for OLMo-1B-DPO, k=3. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

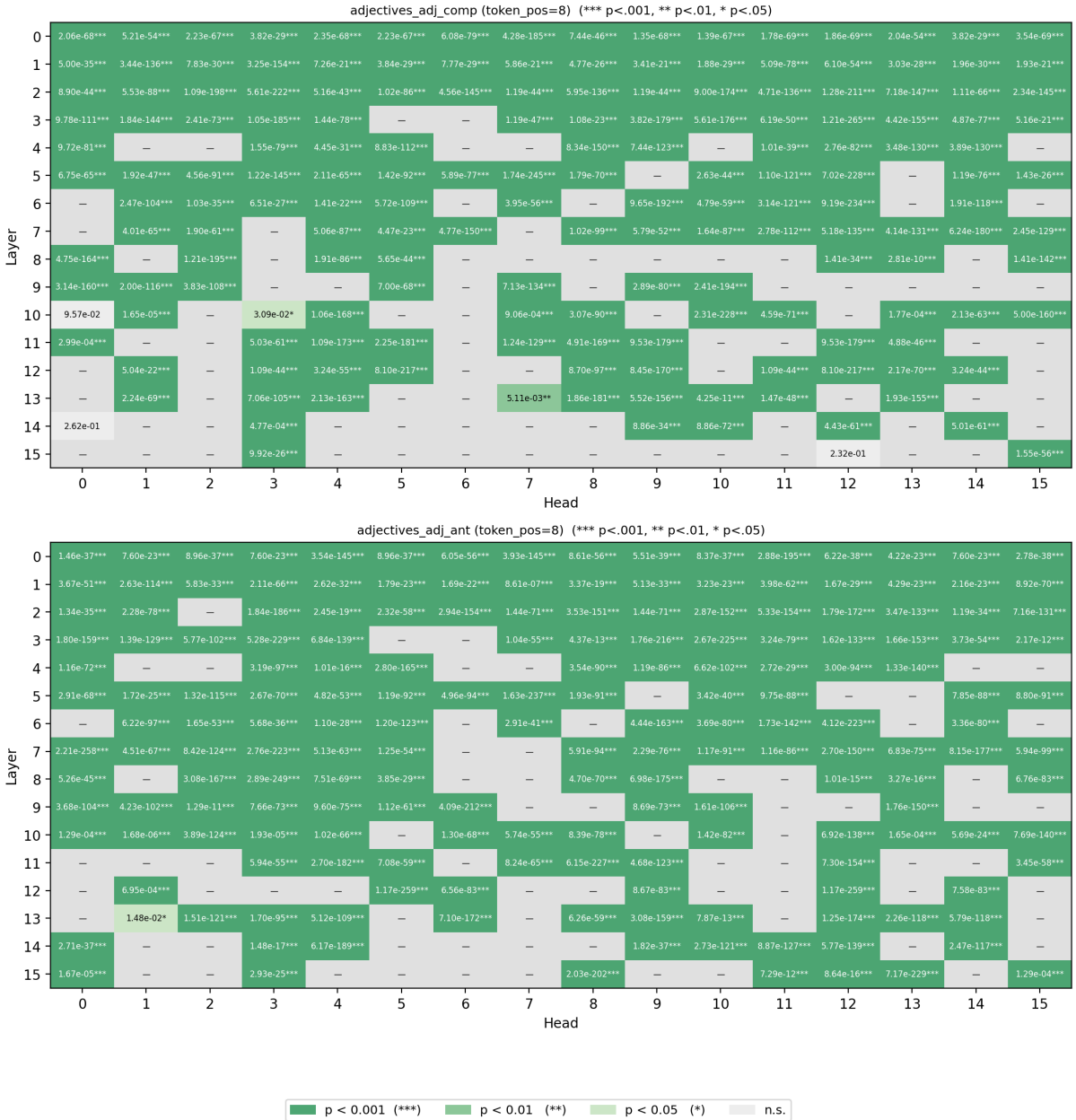

Figure 33: Raw p-values of attention head activity for OLMo-1B, k=1.

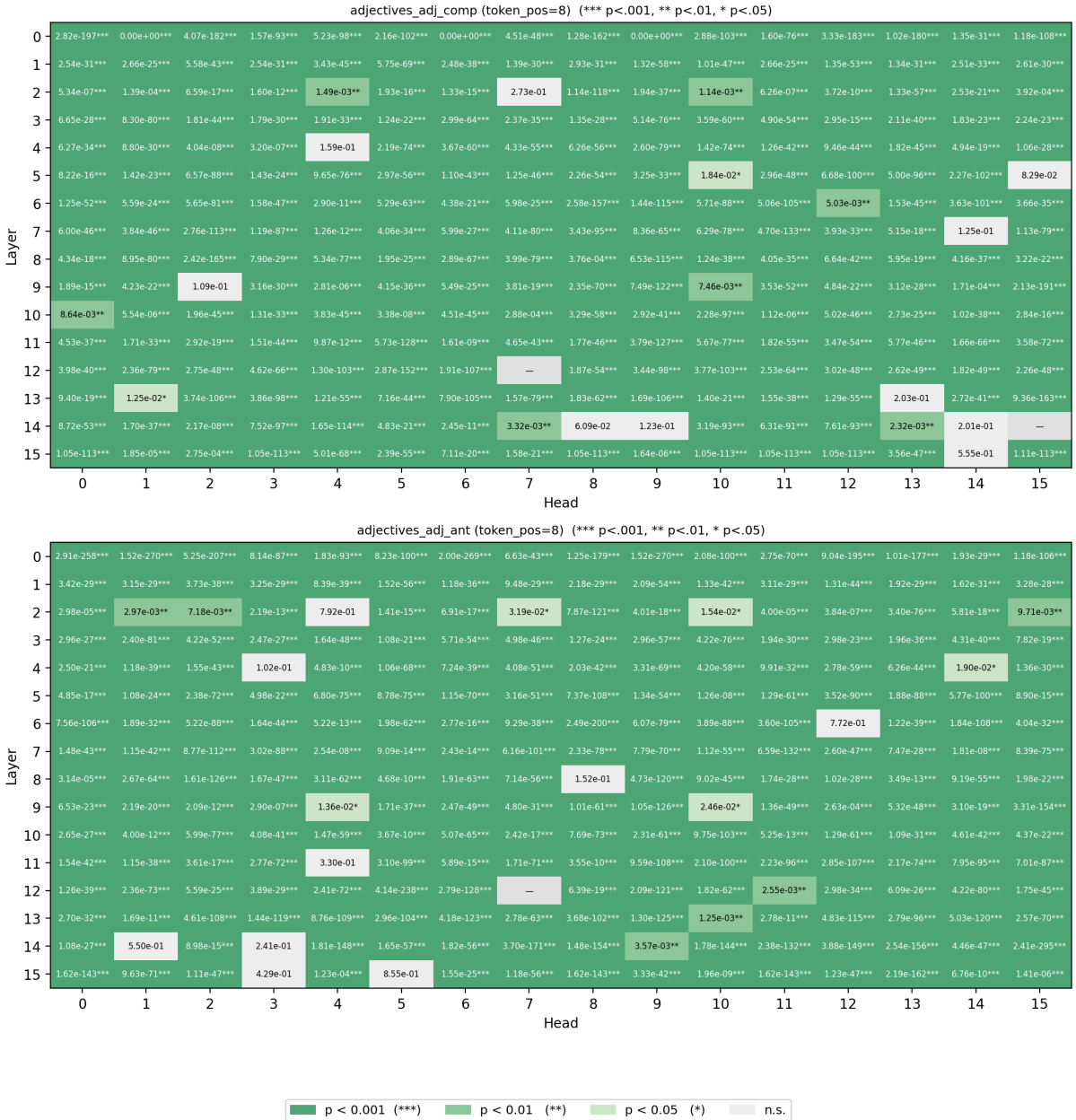

Figure 34: Raw p-values of attention head activity for OLMo-1B, k=2.

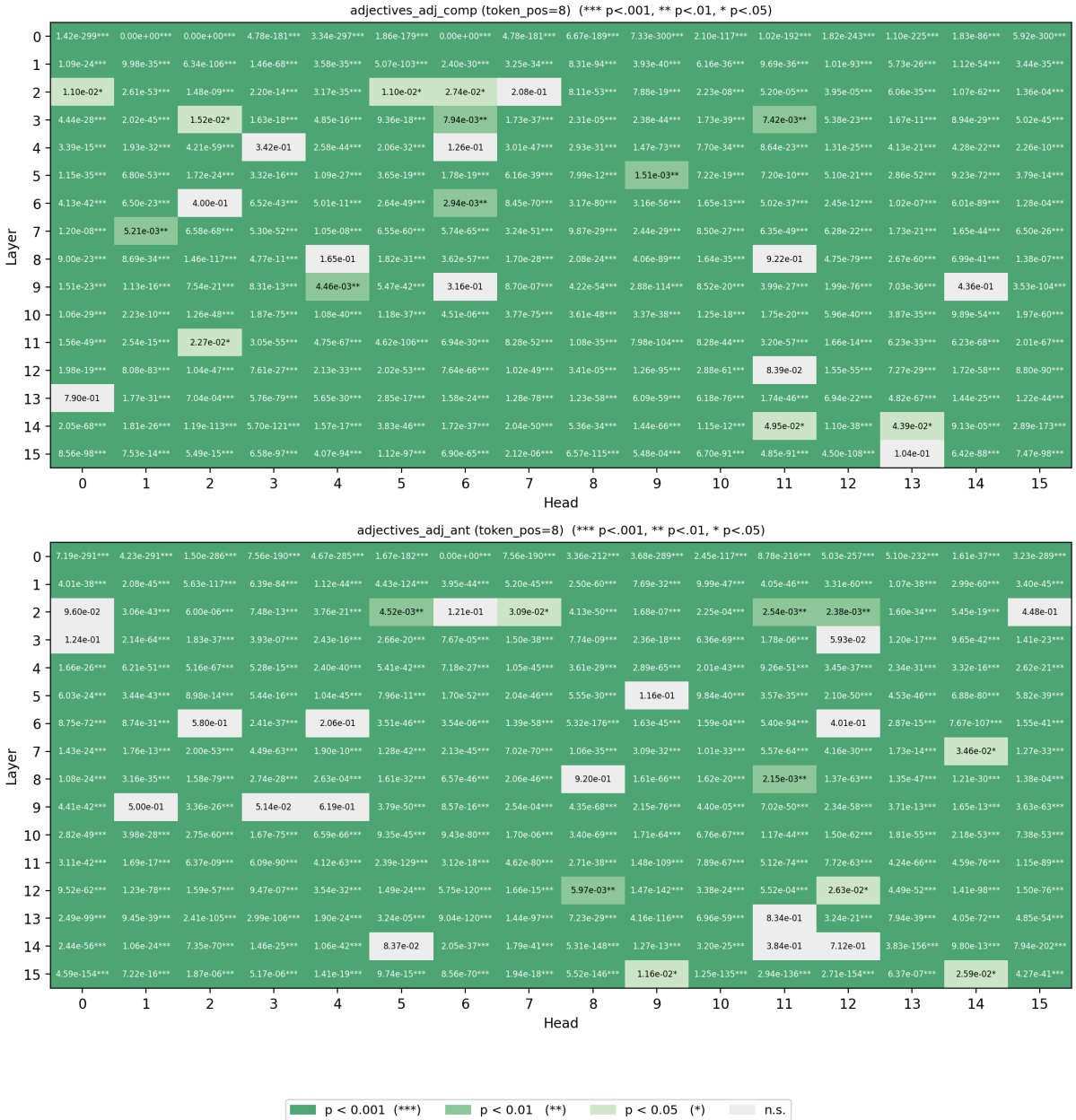

Figure 35: Raw p-values of attention head activity for OLMo-1B, k=3.

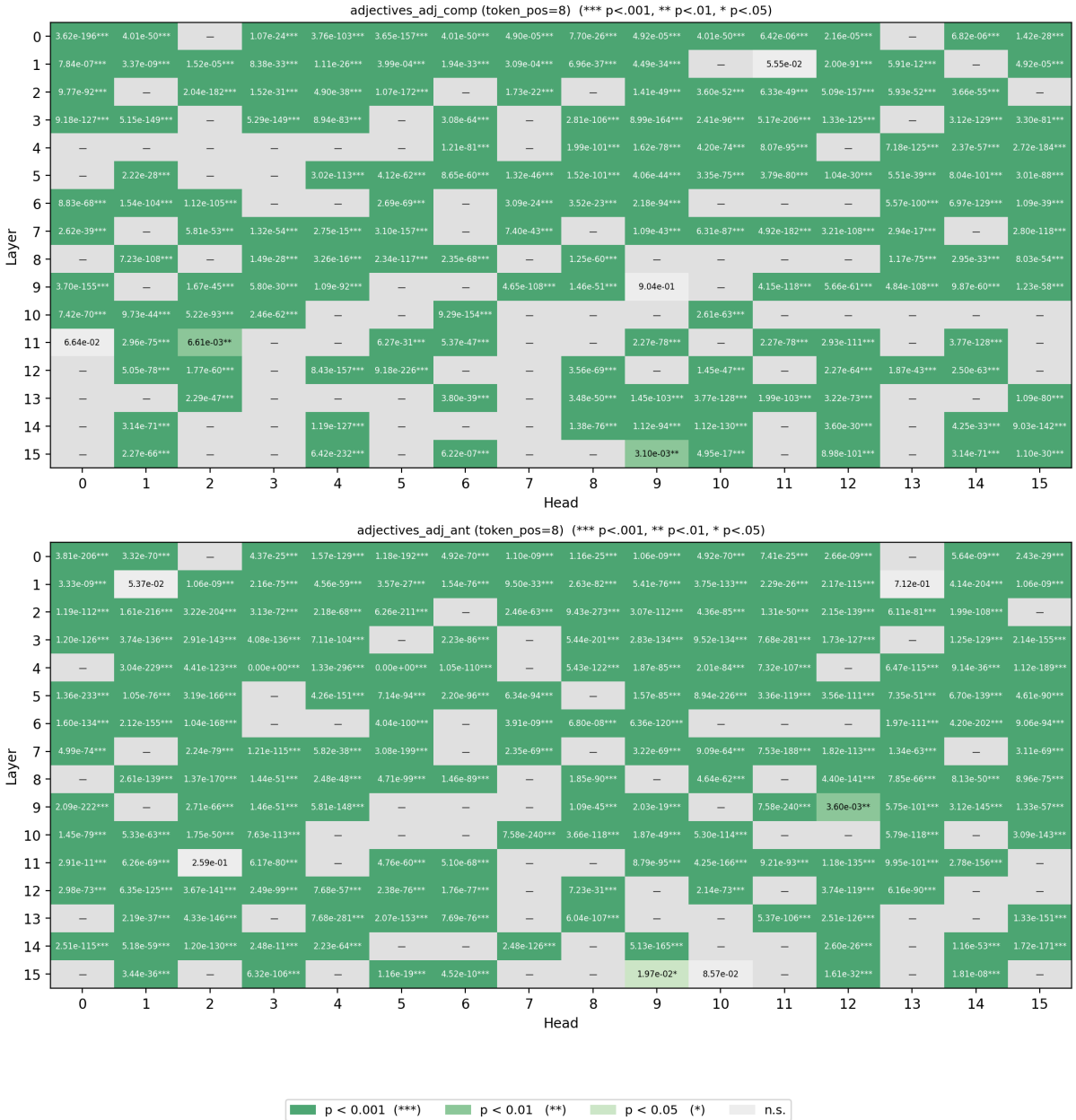

Figure 36: Raw p-values of attention head activity for OLMo-1B-DPO, k=1.

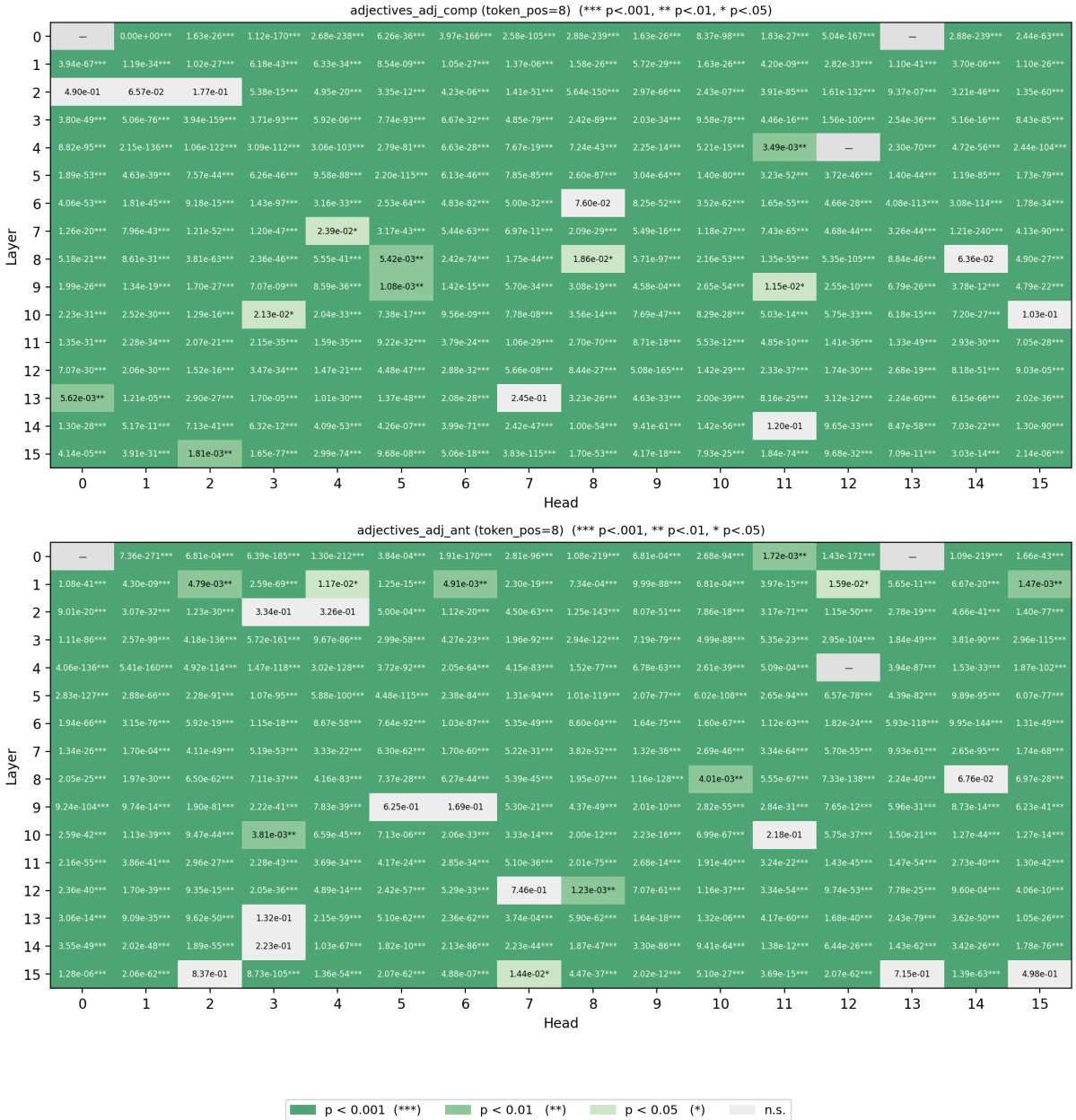

Figure 37: Raw p-values of attention head activity for OLMo-1B-DPO, k=2.

Figure 38: Raw p-values of attention head activity for OLMo-1B-DPO, k=3.

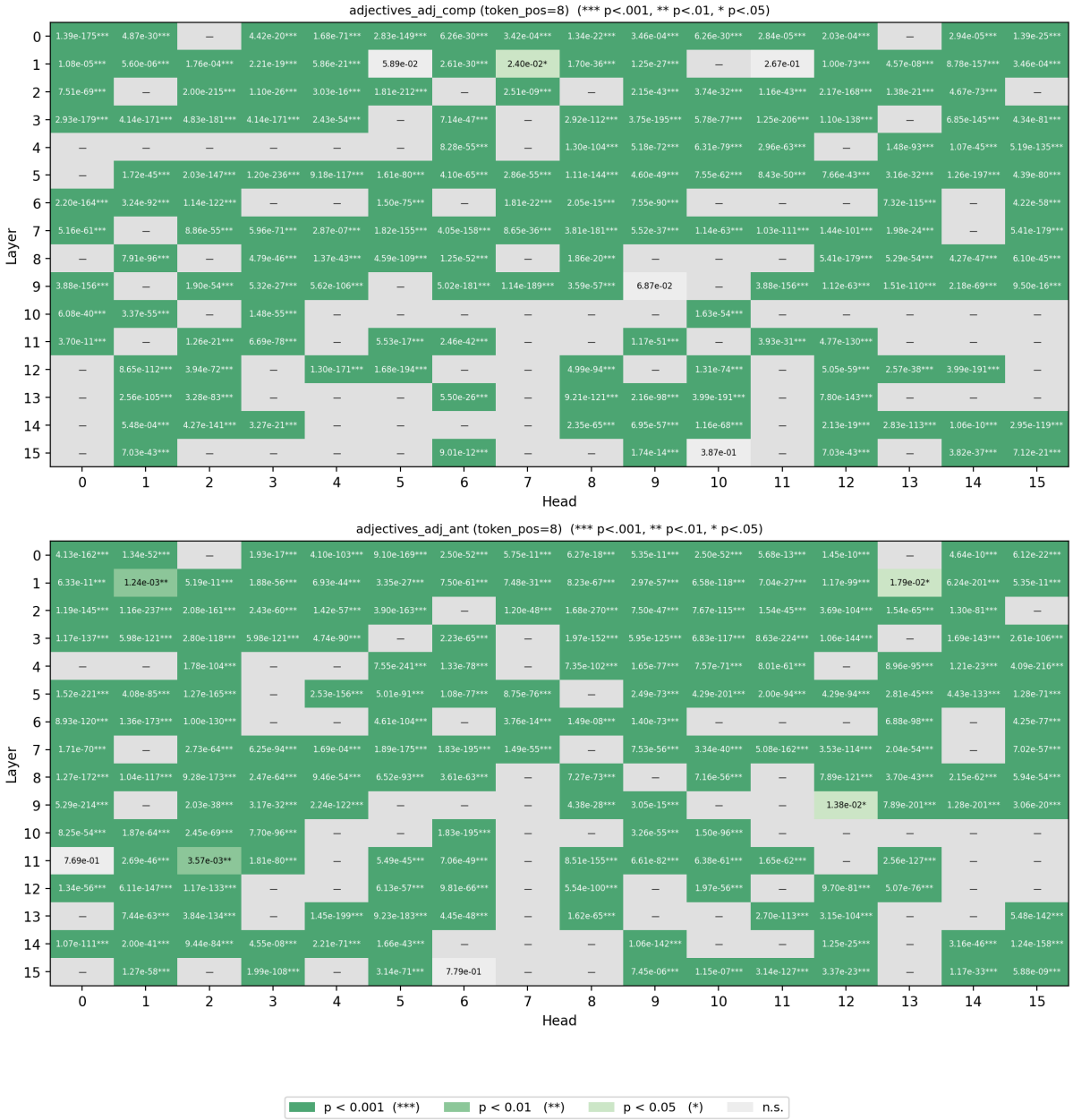

Figure 39: Raw p-values of attention head activity for OLMo-1B-SFT, k=1.

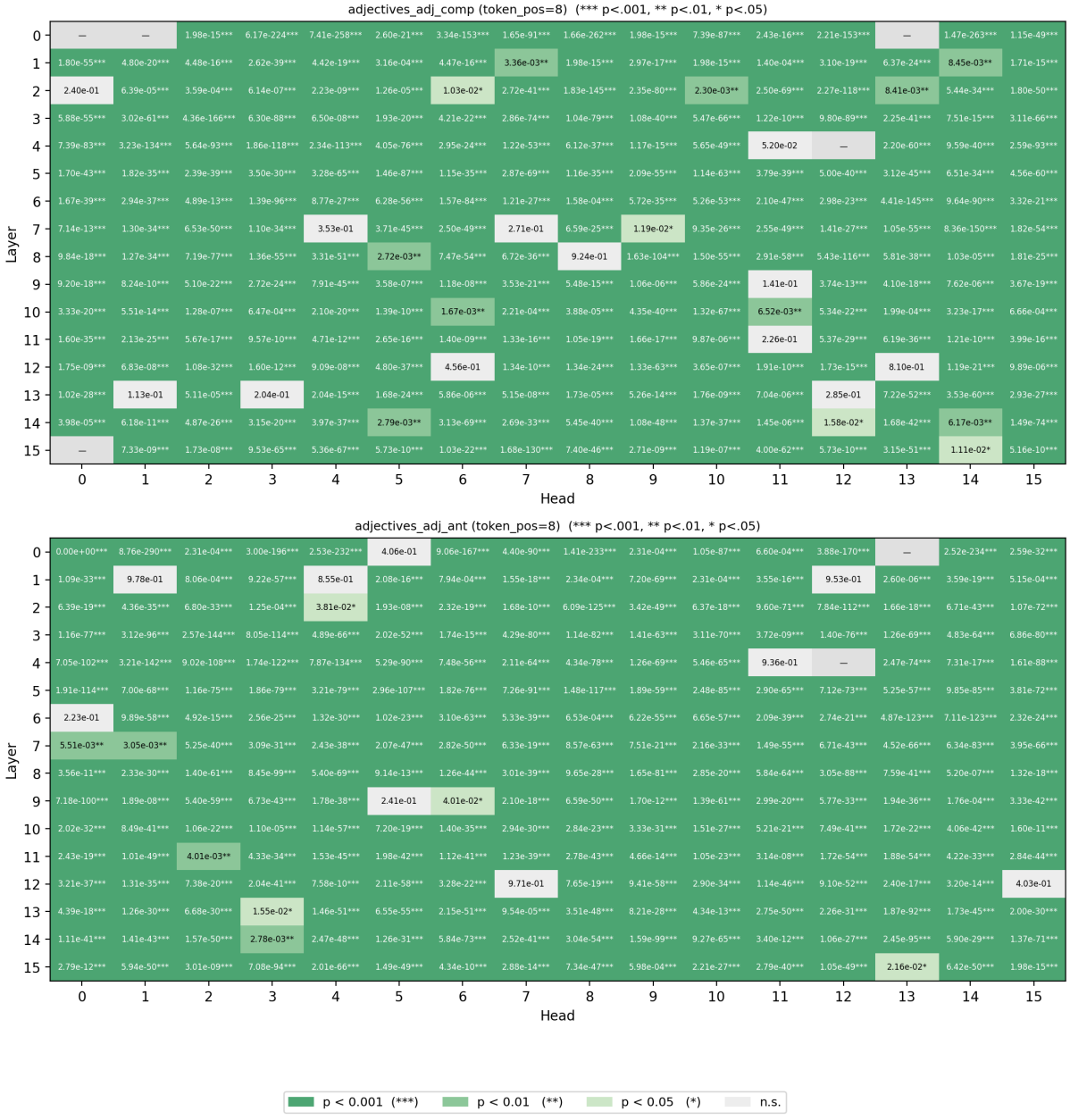

Figure 40: Raw p-values of attention head activity for OLMo-1B-SFT, k=2.

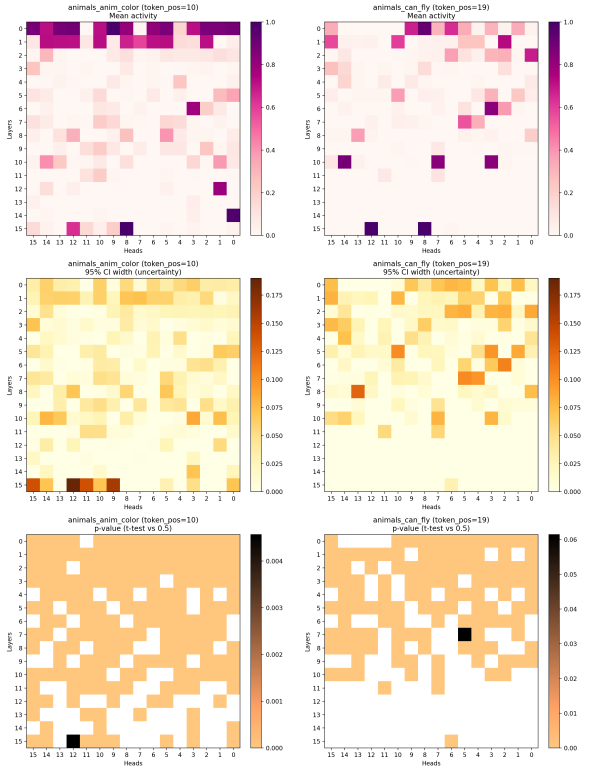

Figure 41: Results of our statistical tests of attention head activity, for OLMo-1B, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

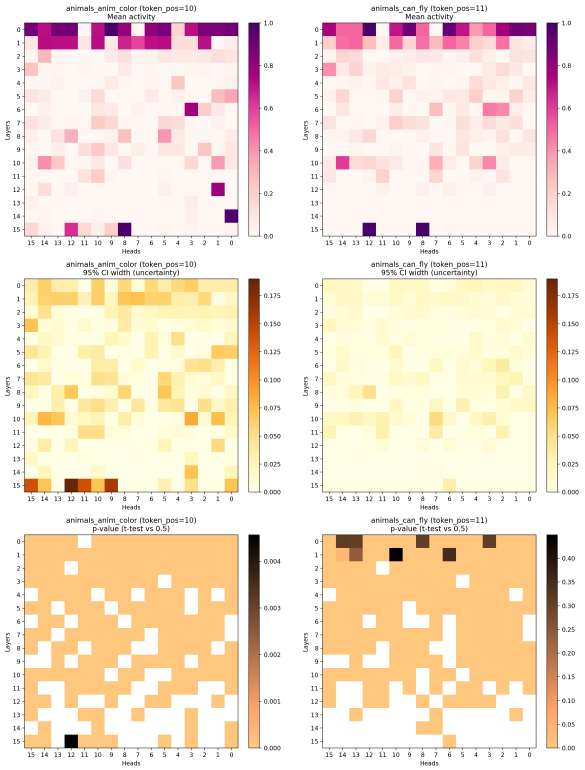

Figure 42: Results of our statistical tests of attention head activity, for OLMo-1B, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value. **Note that the comparison is between the T_{inst} token of ANIMALS: COLOR and an earlier token of ANIMALS: CAN_FLY that represents the first complete "sub-instruction".**

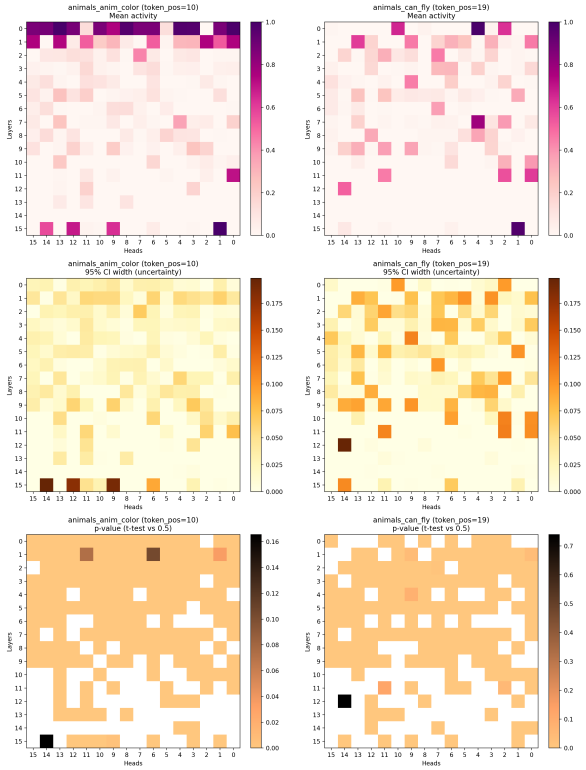

Figure 43: Results of our statistical tests of attention head activity, for OLMo-1B-SFT, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

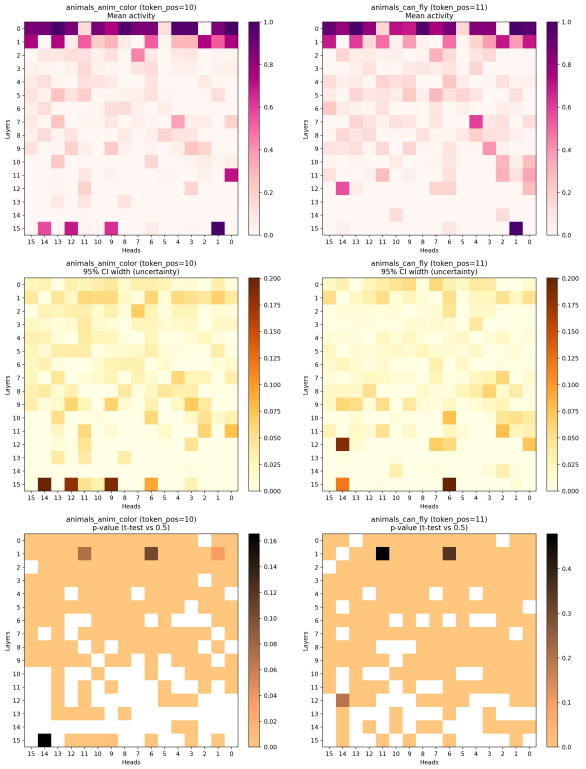

Figure 44: Results of our statistical tests of attention head activity, for OLMo-1B-SFT, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value. **Note that the comparison is between the T_{inst} token of ANIMALS: COLOR and an earlier token of ANIMALS: CAN_FLY that represents the first complete "sub-instruction".**

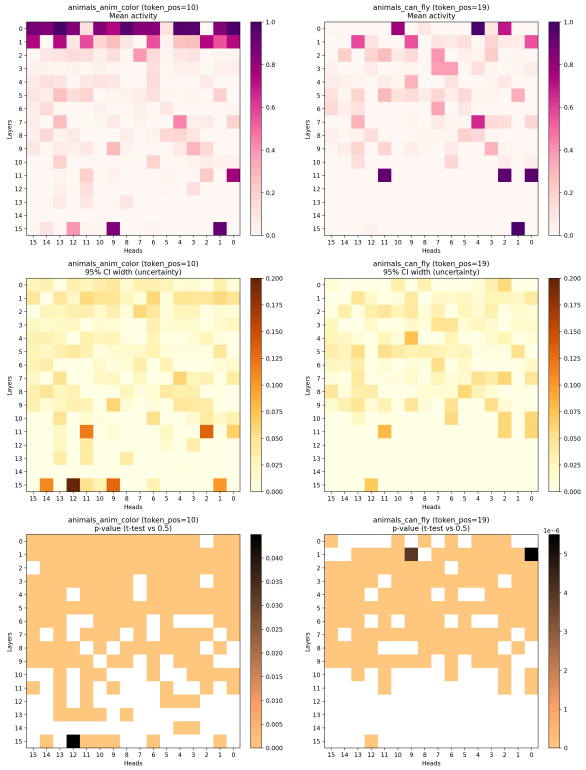

Figure 45: Results of our statistical tests of attention head activity, for OLMo-1B-DPO, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value.

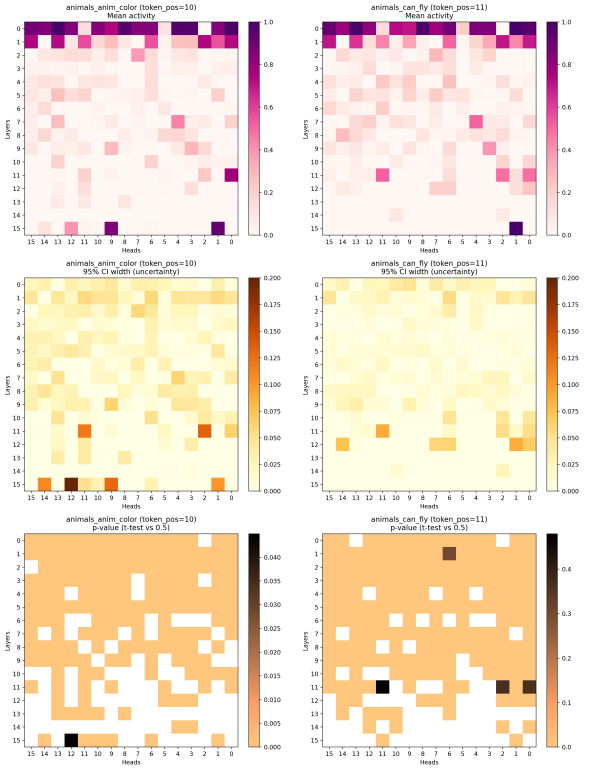

Figure 46: Results of our statistical tests of attention head activity, for OLMo-1B-DPO, k=1. Top: average activity across 100 samples. Middle: boostrapped confidence intervals, with lighter=narrower. Bottom: p-values showing significance of head variance against Gaussian noise, with lighter=lower p-value. **Note that the comparison is between the T_{inst} token of ANIMALS: COLOR and an earlier token of ANIMALS: CAN_FLY that represents the first complete "sub-instruction".**