# StructLens: A Structural Lens for Language Models via Maximum Spanning Trees

**Haruki Sakajo** **Frederikus Hudi** **Yusuke Sakai**

**Hidetaka Kamigaito** **Taro Watanabe**

Nara Institute of Science and Technology (NAIST)
sakajo.haruki.sd9@naist.ac.jp
{frederikus.hudi.fe7,sakai.yusuke.sr9,kamigaito.h,taro}@is.naist.jp

## Abstract

Language exhibits inherent structures, a property that explains both language acquisition and language change. Given this characteristic, we expect language models to manifest their own internal structures as well. While interpretability research has investigated how models compute representations mechanistically through attention patterns and Sparse Autoencoders, the organization of the resulting representations is overlooked. To address this gap, we introduce StructLens, a framework to analyze representations through a holistic structural view. StructLens constructs maximum spanning trees based on the semantic representations in residual streams, inspired by tree representation in dependency parsing, and provides summaries of token relationships in representation space. We analyze how contiguous tokens are also nearby in representation space and find that middle layers show the strongest local-span organization. Moreover, analysis of pre-training checkpoints reveals that smaller local units become detectable earlier in pre-training, and larger units later. Our findings demonstrate that StructLens provides insights into how models organize token representations across layers and training. Our code is available at https://github.com/naist-nlp/structlens.

## 1 Introduction

Language possesses structure. Linguistic phenomena, such as language acquisition and language change, have been explained through underlying structural frameworks [1–4]. Given language's structural nature, we expect that language models (LMs), which are designed to computationally model language, should similarly exhibit their own structural properties [5].

While language exhibits such structural properties, research on LMs, e.g., interpretability and pruning, has frequently overlooked these structures. For example, existing interpretability tools primarily analyze individual tokens or features, e.g., logit lens [6] and Sparse Autoencoders (SAEs) [7]. Similarly, cosine similarity that is employed for inter-layer analysis [8, 9] is fundamentally based on token-to-token comparisons at corresponding positions, making it challenging to capture the holistic structural pattern formed within specific layers. To facilitate global analysis of representations at each layer, approaches that incorporate inter-token relationships and provide comprehensive structural insights are expected to make valuable contributions to LM analysis.

Several studies have utilized parsing techniques developed in Natural Language Processing (NLP) to conduct inter-layer analysis based on inter-token relationships from a linguistic, particularly generative linguistic, perspective. These investigations have demonstrated that attention weights

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

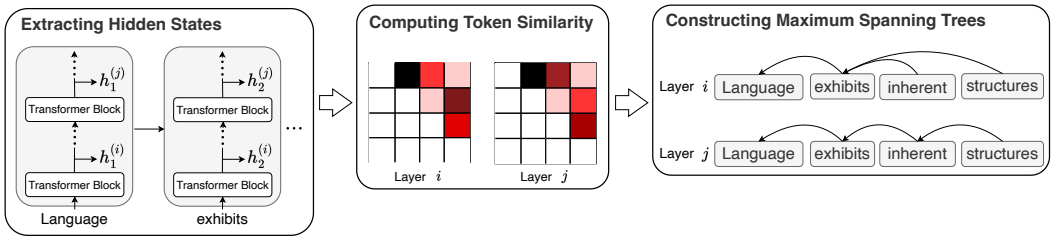



Figure 1: Overview of STRUCTLENS. STRUCTLENS extracts hidden states and then computes inter-token similarity. Finally, it constructs Maximum Spanning Trees using the adjacency similarity matrices.

reflect syntactic structures [10–13], representations encode syntactic information [14–18], and the syntactic structures emerge in a bottom-up manner [19]. Although these studies have revealed that LMs possess and utilize structures, their focus has centered on static, generative grammatical structures that presuppose a certain ground truth structure synthesized by linguists. However, given that language exhibits dynamic structures [2–4] formed through bottom-up processes and LMs are unable to introspect their internal mechanisms, the approaches of bottom-up construction and analysis should be more appropriate to assess the LMs' own internal structure.

To address this gap, we propose STRUCTLENS, a framework that constructs Maximum Spanning Trees (MSTs), i.e., a tree structure connecting all the nodes in a graph with the maximum total edge weight, using LM internal representations, analogous to those employed in dependency parsing in the NLP field [20–23]. Although graph structures are more general, arbitrary complex structures are often difficult to interpret. On the other hand, tree structures provide interpretable representations of high-dimensional data [24]. We therefore leverage tree structures to analyze LM representations. Our approach analyzes residual streams at each layer's output, computing L2 distance between token semantic representations to construct an MST at each layer (see Figure 1). STRUCTLENS provides an order-constrained tree summary of token relationships in representation space. To analyze the trees, we introduce the local-span connectivity metrics that quantify whether connected tree fragments are concentrated on contiguous token intervals. STRUCTLENS finds that LMs exhibit a middle-layer local-span signature, where smaller local spans spread across tokens earlier in pre-training before consolidating into denser, larger-span structures during later steps of pre-training. We also provide a pilot study showing that structure-aware metrics computed over STRUCTLENS improve layer pruning in some settings. Our findings highlight that structure-aware perspectives are potentially useful for LM analysis and optimization.

## 2 Background and Related Work

**Structures in language.** In the study of language, researchers have assumed that language possesses structures, conceived as either static, e.g., generative grammar [1], or dynamic, e.g., usage-based theory [2, 3]. Traditional generative grammar, i.e., transformational grammar, assumes formal rules, while usage-based approaches hypothesize that instances of use shape language representations, allowing analysis of gradual language change.

**Residual stream.** The Transformer [25] updates internal representations gradually by utilizing residual connections. This work assumes a variant of the Transformer with pre-layer normalization architecture [26], which forms a *residual stream* [27]. Formally, given the input features of length $n$, let $d$ be a hidden dimension and $f_\theta^{(\ell)}(\cdot) : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ be the Transformer block at layer $\ell$ that comprises Multi-Head Attention and Multi-Layer Perceptron blocks. The hidden state of the input at the $\ell$-th layer is referred to as the residual stream $\boldsymbol{H}^{(\ell)} \in \mathbb{R}^{n\times d}$, defined as follows:

$$\boldsymbol{H}^{(\ell+1)} = \boldsymbol{H}^{(\ell)} + f_\theta^{(\ell+1)}(\boldsymbol{H}^{(\ell)}). \tag{1}$$

Residual streams in LMs have provided insights into both interpretability work [28–31] and layer pruning methods [8, 32].

**Mechanistic interpretability for language models.** Mechanistic interpretability is an interpretable framework, employing bottom-up methods to reveal models' computational processes and behavior [33]. Research on LM interpretability has examined both activations on the residual stream and the modules that transform them (e.g., Multi-Head Attention, Multi-Layer Perceptron), uncovering the nature of encoded information and functions [34–37]. Research on mechanistic interpretability has also identified models' computational circuits, which reflect underlying behaviors of LMs [38–40]. Logit lens [6] is a technique to analyze intermediate states by projecting intermediate representations into a vocabulary space through the final prediction layer. Previous studies trained probes and evaluated whether targeted information (e.g., syntactic trees) is encoded in representations [14, 15, 41–44]. Sparse Autoencoders (SAEs) are used to identify interpretable features and causal circuits within models, addressing the challenge of superposition, where representations exhibit polysemantic properties [7, 44, 45]. Building on these approaches, we focus on inter-token relationships within individual layers and construct tree structures for each layer, enabling us to provide global views beyond token-level interpretations.

## 3 Method

The primary objective of StructLens is to analyze Transformer layers in terms of the organization of token representations in the representation space from a holistic perspective. Given the residual stream representations of all tokens in an input sequence, we first construct a complete graph whose edge weights reflect pairwise representation similarity. Since arbitrary complex structures of the graph are often hard to analyze and the edge weights vary across layers (see Appendix A), we derive a different graph, i.e., a Maximum Spanning Tree (MST), from each layer of the Transformer. The resulting MST can be interpreted as a structure of the layer-wise representation geometry and enables comparison of the structures across layers.

Formally, let $\boldsymbol{h}_i^{(\ell)}$ denote the residual stream of the $i$-th token immediately after layer $\ell$. For the layer $\ell$, given an input token sequence $\boldsymbol{x}$ of length $n$, we first define a pairwise similarity matrix $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{n \times n}$ as:

$$W_{i,j}^{(\ell)} = \begin{cases} \dfrac{1}{1 + \|\boldsymbol{h}_i^{(\ell)} - \boldsymbol{h}_j^{(\ell)}\|}, & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

This matrix is symmetric and defines a complete graph $\mathcal{G}$ without self-loops, where each node corresponds to a token in $\boldsymbol{x}$ and each edge encodes a relation between two tokens. We use the reciprocal to convert a distance into a similarity for the sake of numerical stability; details are provided in Appendix A.

Given that tokens have a canonical order and autoregressive LMs have left-to-right information flow, we define a forward-constrained adjacency matrix $\boldsymbol{A}^{(\ell)} \in \mathbb{R}^{n \times n}$ as:

$$A_{i,j}^{(\ell)} = \begin{cases} W_{i,j}^{(\ell)} & \text{if } i < j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The forward constraint has two useful consequences, formalized below: it preserves the undirected pairwise similarities in $\boldsymbol{W}^{(\ell)}$ because $\boldsymbol{W}^{(\ell)} = \boldsymbol{A}^{(\ell)} + \boldsymbol{A}^{(\ell)\top}$, and it makes the directed MST equivalent to selecting the most similar preceding token for each non-root token under a unique-predecessor assumption.

**Proposition 3.1.** *Under the forward constraint, StructLens selects the most similar predecessor for each non-root token.*

*Proof.* Under the forward constraint, each non-root token $i$ chooses one preceding token $j < i$ as its parent. Thus, any such parent selection forms a single-root spanning tree.

Let $Pa(i)$ denote the parent token of token $i$. Since the total weight of the tree is $\sum_{i=2}^{n} \boldsymbol{A}_{Pa(i),i}$, and the choice of $Pa(i)$ is independent of the choice of $Pa(k)$ for $k \neq i$, this sum is maximized independently for each token $i$. Thus, $Pa(i) \in \arg\max_{j<i} \boldsymbol{A}_{j,i}$.

Therefore, the directed MST under the forward constraint is equivalent to selecting the most similar preceding token for each non-root token. □

Under the forward constraint, StructLens has a particularly simple interpretation. Each token selects the most similar predecessor in representation space, producing an acyclic directed graph. This makes StructLens scalable and easy to interpret.

For each layer, we build a single-root, order-respecting MST on $\boldsymbol{A}^{(\ell)}$. Although the optimal tree can be obtained by selecting the highest-weight predecessor, we use the algorithm introduced by [46], which runs in $\mathcal{O}(n^2)$ time for a dense graph and is based on Chu-Liu/Edmonds' algorithm [47, 48], as it provides a unified implementation for the present setting and for future StructLens variants with other edge constraints.

StructLens summarizes the geometry of token residual-stream representations as an order-constrained directed tree. Each edge connects a token to a similar preceding token, yielding an interpretable backbone of local and non-local representational relationships within a layer. For simplicity, we refer to this tree as an MST.

## 4 Analyzing Layers through StructLens

### 4.1 Local-span Connectivity in Pretrained Language Models

Inspired by context-dependent units seen in linguistic structures for humans, i.e., chunks, we study whether StructLens trees exhibit local-span connectivity, in which connected subtrees have token indices that form contiguous intervals. We use this as an operational measure of local organization in representation space. Specifically, we focus on the $k$-node subtrees spanning over contiguous position tokens without any skips, as shown in Figure 2, and we refer to them as *contiguous subtrees*.

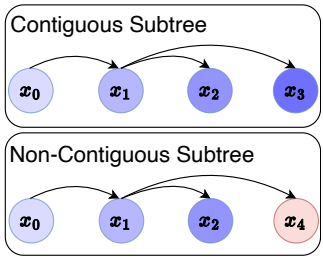



Figure 2: Contiguous and non-contiguous 4-node subtrees.

We measure the *contiguous subtree ratio*, which quantifies the density of contiguous subtrees, indicating how strongly the $k$-node subtrees concentrate on local token spans, and the *contiguous token ratio*, which measures the coverage of such tokens. These ratios characterize token relationship trees in representation space with respect to local connectivity.

Formally, a $k$-node subtree is a connected induced subgraph of a tree $\mathcal{T}$ containing $k$ nodes. A contiguous $k$-node subtree is a $k$-node subtree whose token indices form an interval $\{a, a+1, \ldots, a+k-1\}$. Let $C^k_{cont}(\mathcal{T})$ and $C^k_{all}(\mathcal{T})$ denote the number of contiguous $k$-node subtrees of a tree $\mathcal{T}$ and the total number of $k$-node subtrees of $\mathcal{T}$, respectively. The *contiguous subtree ratio* of $\mathcal{T}$ is defined as: $\frac{C^k_{cont}(\mathcal{T})}{C^k_{all}(\mathcal{T})}$. We also investigate the number of tokens covered by contiguous subtrees. Let $\mathcal{S}_k(\mathcal{T})$ be the set of tokens in the contiguous $k$-node subtrees of $\mathcal{T}$. Let $\mathcal{N}_k(\mathcal{T}) = \bigcup_{S \in \mathcal{S}_k(\mathcal{T})} V(S)$ be the set of node positions appearing in any such subtree, where $V(S)$ denotes the node set of subtree $S$. The *contiguous token ratio* is defined as: $\frac{|\mathcal{N}_k(\mathcal{T})|}{n}$.

To illustrate the difference between the contiguous subtree ratio and the contiguous token ratio, we consider 4-node contiguous subtrees in three simple 7-node trees, as illustrated in Figure 3. We present rooted trees by strict S-expressions, where the first element is the parent and the following elements are its children. For the chain tree $(1(2(3(4(5(6(7)))))))$, the contiguous subtree ratio and the contiguous token ratio are 1. For the star tree $(1(2)(3)(4)(5)(6)(7))$, the total number of 4-node subtrees is 20, the total number of contiguous subtrees is 1, and the total number of contiguous tokens is 4. Therefore, the contiguous subtree ratio is $\frac{1}{20}$ and the contiguous token ratio is $\frac{4}{7}$. For the binary tree $(1(2(4)(5))(3(6)(7)))$, the total number of 4-node subtrees is 6, and the total number of contiguous subtrees is 1. Therefore, the contiguous subtree ratio is $\frac{1}{6}$ and the contiguous token ratio is $\frac{4}{7}$.

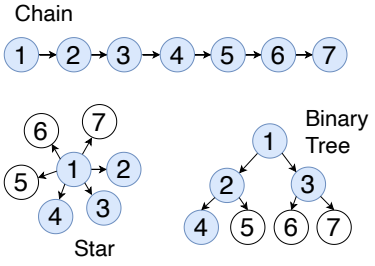



Figure 3: Illustration of trees. Colored nodes form contiguous 4-node subtrees.

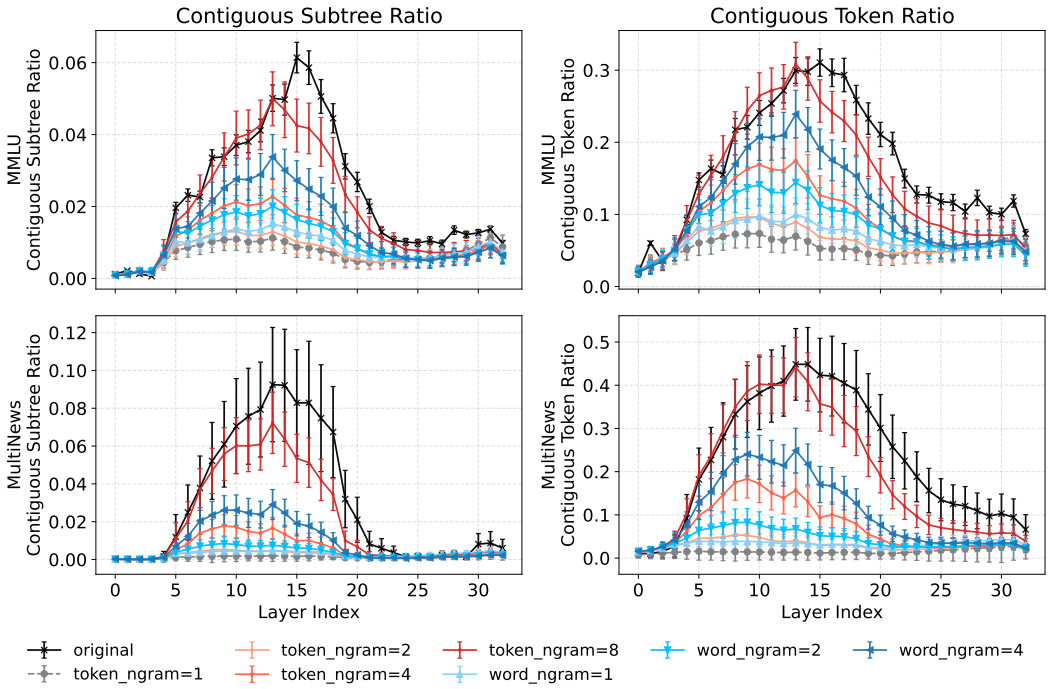



Figure 4: Visualization of the layer-wise evolution of the contiguous 4-node subtrees and tokens in them for the original input order and the shuffled one for Llama3.1 8B. The x-axis denotes layer depth, and the y-axis denotes the contiguous subtree ratio and the contiguous token ratio.

Furthermore, to determine whether contiguous subtrees are derived from implicit positional information or patterns dependent on natural token order, we compare the original input order with the shuffled input order. We also shuffle input tokens at the token or word n-gram levels to examine whether models consider or ignore word boundaries and semantic coherence. If LMs exhibit contiguous subtrees only when the input is ordered as in the original order, this indicates that they are sensitive to natural token order beyond the tested shuffled baseline.

**Experimental settings.** We employ Llama3.1 8B [49] and Qwen2.5 7B [50] for our experiments. The evaluation datasets are MMLU [51], which is an English multiple-choice Question-Answering dataset with four choices, and MultiNews [52], which is a summarization dataset in English. We randomly sample instances from each dataset and employ prompt templates with five-shot examples from the development set of each dataset, as used in the MMLU paper and LongBench [53] for MultiNews. As a case study, we investigate 4-node contiguous subtrees. We also test 3- and 5-node contiguous subtrees on MMLU. A detailed description of the experimental settings is provided in Appendix B. When shuffling input tokens, we keep the position of the begin-of-sentence token and shuffle other tokens. We run the shuffling experiment with three seeds {0, 1, 2} and report the average.

**Results.** Figure 4 shows the contiguous subtree and token ratios for all layers by averaging over the samples. This figure reveals that, from lower to middle layers, i.e., 0% to 50% of the layer percentage, both ratios become larger, indicating that contiguous local structures become more frequent and more widely distributed across tokens. In higher layers, the ratios decrease, suggesting that this local organization is reduced. This pattern, where the intermediate layers behave differently from other layers, has also been observed in geometric metrics and embedding tasks [54, 55], supporting that the structures derived by StructLens reflect meaningful properties of layer-wise representations.We also conducted the same experiments on MMLU and multinews with the original input order for 3-node and 5-node subtrees for both Llama3.1 8B and Qwen2.5 7B in Appendix C.1, which show similar trends to those observed for 4-node subtrees. This indicates that this middle-layer pattern is not only for 4-node subtrees but also for another local-span pattern. Shuffling results show that

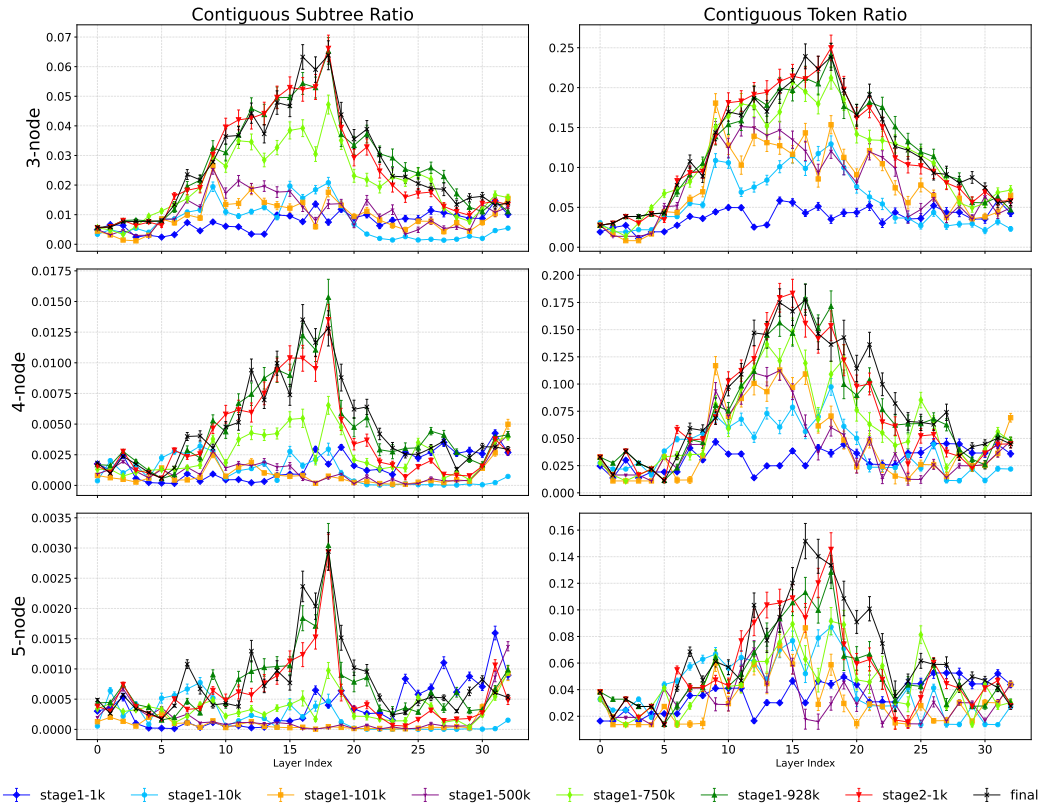



Figure 5: Visualization of the layer-wise evolution of the 3-, 4-, and 5-node contiguous subtrees and tokens in them across multiple checkpoints of OLMo2 7B for MMLU. In each plot, the x-axis denotes the layer index, and the y-axis denotes the contiguous subtree and token ratio.

models organize fewer contiguous subtrees at the smaller unit levels, indicating that the contiguous subtrees observed in the original setting depend on the original token order. These results suggest that LMs organize token representations into contiguous structures in intermediate layers before distributing them or forming non-local structures in higher layers.

### 4.2 Training Dynamics through StructLens

During pre-training, downstream task performance improves gradually, while mid-training yields substantial performance gains across several tasks [56, 57]. We further examine when LMs acquire the behavior in the middle layers during pre-training and mid-training by measuring contiguous subtree and token ratios across multiple training checkpoints, revealing how and when they do so.

**Experimental settings.** We employ OLMo2 7B [57] and its checkpoints during pre-training. We use checkpoints at steps {1,000, 10,000, 101,000, 500,000, 750,000, 928,646} of stage 1, and at 1,000 steps of one of the checkpoints of stage 2 called Mid-training [57]. We refer to them as stage1-1k, stage1-10k, stage1-101k, stage1-500k, stage1-750k, stage1-928k, and stage2-1k, respectively. The evaluation dataset is MMLU, and we compute the contiguous subtree and token ratios for the 3-, 4-, and 5-node subtrees, and test for the original input order and the shuffled order. We use the same settings as reported in Section 4.1.

**Results.** Figure 5 shows the contiguous subtree and token ratios for multiple checkpoints as reported in Section 4.1. The shuffling results are reported in Appendix C.2. This figure shows that the subtree ratio in the middle layers differs between later checkpoints (stage1-928k and stage2-1k) and earlier ones, while the model behaves similarly across checkpoints in the lower layers. By analyzing different subtree sizes $k \in \{3, 4, 5\}$, StructLens reveals a scale-dependent emergence of local-span connectivity. Smaller subtrees become detectable earlier in training, especially in the

token-coverage metric, whereas larger contiguous subtrees emerge later in the middle layers. This suggests that the model first develops short-range local connectivity, and later training consolidates this connectivity into denser, larger-span tree fragments. The contiguous token ratio also suggests that such structures are distributed broadly across tokens at earlier steps. Notably, this emergence is not observed in training, optimizer, or downstream-task metrics[1]. Comparing the structural signal with loss, optimizer, and downstream metrics can be an important direction of future work.

### 4.3 Token Organization in Language Models through StructLens

Layer-wise analysis shows that the contiguous local structures become more dominant in the middle layers and are reduced in the higher layers. Training dynamics further show that this behavior emerges at later checkpoints. The contiguous token ratio increases during earlier steps of pre-training, indicating that contiguous local structures become more broadly distributed across tokens. In contrast, the contiguous subtree ratio exhibits a sharper increase at later checkpoints, indicating that such structures become more dominant among tree fragments. These trends suggest a two-stage process during pre-training: first, smaller contiguous structures spread across the sequence, and later, they become more concentrated and larger within the representation geometry. In summary, StructLens reveals how token representations are organized across layers and during pre-training.

## 5 Inter-Layer Similarity through StructLens

### 5.1 Measuring Inter-Layer Similarity through StructLens

For analyzing layer redundancy in LMs, established methods, e.g., cosine similarity, are employed to quantify layer similarity [8, 9]. These conventional approaches measure similarity between representations at corresponding positions, capturing local pairwise relationships. However, these methods lack a global perspective encompassing intra-layer token relationships and do not provide a holistic view of layer-level interaction. In this study, we compute layer similarity with three structure-aware similarity metrics to measure comprehensive and holistic relationships using StructLens.

**Cosine Similarity for StructLens (Cos-Struct).** For each subtree of depth 2, we compute the average of the hidden representations of the parent and its children, yielding a flattened subtree of depth 1. This process is applied recursively until only a single representation remains at the root node. Let $\mathbb{K}_i$ be the set of child nodes for $i$, defined as $\mathbb{K}_i = \{\, j \in \{1, \ldots, n\} \mid Pa(j) = i \,\}$. The aggregated representation by averaging at node $i$ is defined recursively as:

$$\bar{\boldsymbol{h}}_i = \frac{1}{|\mathbb{K}_i| + 1} \left( \frac{\boldsymbol{h}_i}{\|\boldsymbol{h}_i\|} + \sum\nolimits_{j \in \mathbb{K}_i} \bar{\boldsymbol{h}}_j \right). \tag{4}$$

The aggregated representation at the root node of layer $\ell$ is denoted by $\bar{\boldsymbol{h}}^{(\ell)}$. The structural similarity between two layers, $\ell_a$ and $\ell_b$, is then measured by the cosine similarity of their aggregated root representations:

$$\text{score}_{\text{Cos-Struct}}(\ell_a, \ell_b) = \cos\left( \bar{\boldsymbol{h}}^{(\ell_a)}, \bar{\boldsymbol{h}}^{(\ell_b)} \right). \tag{5}$$

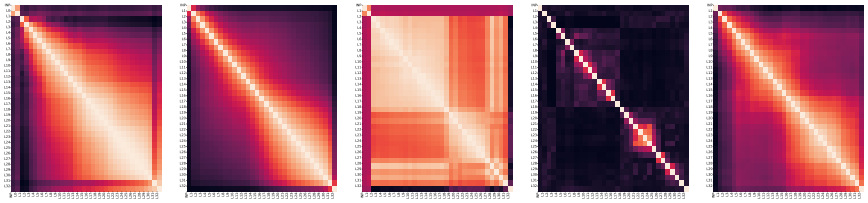
Cos-Struct operates in $O(n)$ time. Although Cos-Struct incorporates structural aggregation, it still does not directly measure the structural similarity between trees induced by StructLens.

**Tree Edit Distance (Tree-Edit).** The Tree Edit Distance [58] has been widely applied and studied as a method for quantifying dissimilarity between two ordered labeled trees [59]. Here, we explore its utility as a negative similarity score.

For a given graph $\mathcal{G}$, let $\mathcal{T}_a$ denote the StructLens maximum spanning tree corresponding to layer $\ell_a$. Let $\mathcal{P}(\mathcal{T}_a, \mathcal{T}_b)$ be the set of edit scripts that transform $\mathcal{T}_a$ into $\mathcal{T}_b$, and let $c(o)$ denote the cost for an edit operation $o$ in an edit script of $\pi$. The Tree-Edit score is defined as:

$$\text{score}_{\text{Tree-Edit}}(\ell_a, \ell_b) = -\left( \min_{\pi \in \mathcal{P}(\mathcal{T}_a, \mathcal{T}_b)} \sum\nolimits_{o \in \pi} c(o) \right). \tag{6}$$

[1] Available at `https://wandb.ai/ai2-llm/OLMo-2-1124-7B/reports/OLMo-2-7B-Nov-2024--VmlldzoxMDUzMzE1OA`.



(a) CKA (b) Cos-Base (c) Cos-Struct (d) Tree-Edit (e) Edge-Edit

Figure 6: Inter-layer similarity samples of Llama3.1 8B for each metric on MMLU. Bright color represents high similarity, while dark color represents low similarity.

Tree-Edit operates in $O(n^4)$ time. Tree-Edit, however, is unable to move an entire subtree since such changes require deletion and insertion operations recursively for all nodes and edges in the subtree.

**Edge Edit Distance (Edge-Edit).** We employ a more straightforward edge-based edit distance metric, which mitigates score variations caused by the subtree movement between layers, providing a direct and more stable structural comparison. We define the edge-set $\mathbb{S}_{\mathcal{T}}$ of a tree $\mathcal{T}$ as the set of parent-child pairs. Let $r$ be the root node, and let $Pa_T\,(i)$ denote the parent node of any node $i$ other than the root $r$. The edge-set is then given by:

$$\mathbb{S}_{\mathcal{T}} \;=\; \{\,(i, Pa_T(i)) \;\mid\; i \in \{1,\ldots,n\},\; i \neq r\,\}. \tag{7}$$

Let $\mathcal{T}_a$ and $\mathcal{T}_b$ be the spanning trees corresponding to layer $\ell_a$ and $\ell_b$, respectively, and let $\mathbb{S}_{\mathcal{T}_a}$ and $\mathbb{S}_{\mathcal{T}_b}$ be the respective edge-set as defined in Equation 7. As the two trees have the same set of nodes and the same number of edges, the Edge-Edit score equals their edge set difference:

$$\text{score}_{\text{Edge-Edit}}(\ell_a, \ell_b) = -\left(|\mathbb{S}_{\mathcal{T}_a} \backslash \mathbb{S}_{\mathcal{T}_b}| + |\mathbb{S}_{\mathcal{T}_b} \backslash \mathbb{S}_{\mathcal{T}_a}|\right). \tag{8}$$

This metric directly counts edge insertions and deletions, avoids inflated costs from subtree movements, and more stably measures structural similarity across layers in $O(n)$ time.

### 5.2 Non-Structural Metrics

To investigate how structural metrics capture inter-layer similarity, we compare them with non-structural metrics, i.e., Centered Kernel Alignment [60] and Cosine Similarity.

**Centered Kernel Alignment (CKA).** Centered Kernel Alignment (CKA) can be used to compute global inter-layer similarity. We compute CKA using the unbiased estimator of the Hilbert-Schmidt Independence Criterion (HSIC) [61]. Formally, the inter-layer similarity with CKA is defined as:

$$\text{score}_{\text{CKA}}(\ell_a, \ell_b) = \frac{\text{HSIC}(\boldsymbol{K}, \boldsymbol{L})}{\sqrt{\text{HSIC}(\boldsymbol{K}, \boldsymbol{K})\text{HSIC}(\boldsymbol{L}, \boldsymbol{L})}}, \tag{9}$$

where $\boldsymbol{K} = \boldsymbol{H}^{(\ell_a)}\boldsymbol{H}^{(\ell_a)^\top}$ and $\boldsymbol{L} = \boldsymbol{H}^{(\ell_b)}\boldsymbol{H}^{(\ell_b)^\top}$ denote the linear Gram matrices. To mitigate the statistical bias caused by the finite sample size $n$, we employ the unbiased estimator of HSIC [61].

**Cosine similarity (Cos-Base).** Cosine similarity is a widely used metric for comparing vector representations. We measure inter-layer similarity using Cos-Base as:

$$\text{score}_{\text{Cos-Base}}(\ell_a, \ell_b) = \sum\nolimits_i^n \cos\left(\boldsymbol{h}_i^{(\ell_a)}, \boldsymbol{h}_i^{(\ell_b)}\right). \tag{10}$$

### 5.3 Inter-Layer Similarity Analysis

We analyze language models using the similarity of tree structures across layers obtained via StructLens. We apply StructLens to representations of each sampled instance in the datasets and then compute inter-layer similarity using Equations 9, 10, 5, 6, and 8.

**Experimental settings.** We employ Llama3.1 8B [49] and Qwen2.5 7B [50] for our experiments. The evaluation datasets are MMLU [51], which is a multiple-choice Question-Answering dataset with four choices, and its Chinese equivalent, CMMLU [62]. We randomly sample instances and compute inter-layer similarity. Details of experimental settings are provided in Appendix B.

**Results.** The inter-layer similarity for Llama3.1 8B on MMLU is illustrated in Figure 6, where the x-axis and y-axis represent the layer indices. This figure shows that Edge-Edit exhibits diagonal clustering patterns, forming discrete groupings characterized by high inter-layer similarity, which we refer to as islands, and that these patterns are observed with the k-NN metric [63]. We report the similarity for CMMLU and Qwen2.5 7B in Appendix D, where Edge-Edit also exhibits diagonal clustering patterns, and these islands remain consistent across model family and size (see Appendix D). We evaluate the clustering consistency across samples and the clustering quality in Appendix E. The results show that the clusterings produced by these metrics are consistent with $k = 3$ or $k = 4$, except for Tree-Edit. We investigate these islands through frequent subtree mining and logit lens in Appendices F and G, which suggest that islands correspond to meaningful groupings in the LMs.

### 5.4 Layer Pruning through StructLens

As an exploratory application of StructLens, we test whether StructLens-derived layer similarity can provide signals for layer pruning.

**Layer pruning algorithm.** Layer pruning algorithms for Transformer LMs [8, 32, 64] identify and prune layers that produce relatively small modifications to representations based on their similarity across layers. This approach leverages the residual connections in the Transformer as described in Equation 1 and quantifies layer importance to determine layers for removal. We employ the metric introduced in ShortGPT [8] and assess layer influence through StructLens.

**Layer influence.** ShortGPT [8] computes layer influence (importance) using inter-layer cosine similarity, and subsequently removes layers from the model in ascending order of importance. In ShortGPT, the $i$-th layer influence, referred to as Block Influence (BI), is defined using Equation 10 as:

$$\text{CosBase-BI}_i = 1 - \text{score}_{\text{Cos-Base}}(\ell_i, \ell_{i-1}) \tag{11}$$

Additionally, we calculate influence using another non-structural metric, CKA (Equation 9), and three structure-aware StructLens similarity metrics, namely Cos-Struct (Equation 5), Tree-Edit (Equation 6), and Edge-Edit (Equation 8). CKA-BI, CosStruct-BI, TreeEdit-BI, and EdgeEdit-BI are computed similarly to Equation 11 by replacing the Cos-Base score with CKA, Cos-Struct, Tree-Edit, and Edge-Edit, respectively. Since the score ranges for TreeBI and EdgeBI depend on inputs, we normalize them on a per-sample basis, taking into account the theoretical bounds of each metric.

**Experimental settings.** We employ Llama3.1 8B Instruct and Qwen2.5 7B Instruct. For each model, we remove approximately 10% of layers in ascending order of BI scores according to each metric. For evaluation datasets, we use ARC-Easy [65], MMLU, and CMMLU as Question-Answering datasets, GSM8K [66] as a QA dataset with multi-step reasoning, and Multinews and VCSUM [67] as summarization datasets. We measure accuracy for the QA tasks and Rouge-L F1 [68] for summarization tasks. We employ McNemar's test [69] to assess the statistical significance of differences in accuracy, and the paired bootstrap test for ROUGE-L F1. Layer removal calibration utilizes 10 samples from the English Wikipedia dataset [70], as performed in LaCo [32]. Details of experimental settings are provided in Appendix B.

**Results.** The removed layers are presented in Table 1, and we show the layer importance of each layer in Appendix H. The layer-pruning results, as shown in Table 1, indicate that tree-aware metrics outperform non-structural metrics for Llama3.1 8B Instruct, suggesting that structural metrics are helpful for layer pruning in this setting. While each metric shows inconsistent results across datasets for Qwen2.5 7B Instruct, CosStruct-BI exhibits better or comparable performance to the baselines on tasks that require generation, i.e., GSM8K, Multinews, and VCSUM. We also evaluate pre-trained models on multiple QA tasks, and these models exhibit a similar tendency to instruct-tuned models, as shown in Appendix H. These results demonstrate that there is room to improve layer-pruning metrics and that structural metrics potentially provide insights into them.

Table 1: Pruning results. Each value denotes the accuracy or the correctness of an answer in a task, with higher values indicating better performance. Values of each dataset result denoted by † are statistically significant ($p < 0.05$) compared to CosBase-BI.

| Metric | Layers removed | ARC-Easy | MMLU | CMMLU | GSM8K | Multinews | VCSUM | Avg. |
|---|---|---|---|---|---|---|---|---|
| Llama3.1 8B Instruct | | | | | | | | |
| Dense | - | 85.8 | 68.4 | 55.9 | 70.4 | 26.1 | 16.6 | 53.8 |
| CosBase-BI | 24 25 26 27 | 78.3 | 66.8 | 54.4 | 43.0 | 23.1 | 8.7 | 45.7 |
| CKA-BI | 24 25 26 31 | 75.3† | 66.9 | 54.2 | 5.8† | 18.2† | 1.9† | 37.0 |
| CosStruct-BI | 23 24 25 26 | **79.3** | **67.8**† | 55.1† | **57.1**† | **25.1**† | **16.2**† | **50.1** |
| TreeEdit-BI | 23 24 26 27 | 78.1 | 66.9 | **55.8**† | 40.9† | 22.7 | 10.7† | 45.8 |
| EdgeEdit-BI | 23 24 25 26 | **79.3** | **67.8**† | 55.1† | **57.1**† | **25.1**† | **16.2**† | **50.1** |
| Qwen2.5 7B Instruct | | | | | | | | |
| Dense | - | 87.6 | 74.3 | 81.1 | 80.4 | 23.6 | 17.7 | 60.8 |
| CosBase-BI | 15 16 17 | 83.7 | 56.1 | 56.8 | 35.9 | 21.6 | **17.5** | 45.3 |
| CKA-BI | 4 25 27 | 80.1† | **70.0**† | **75.7**† | 26.3† | 12.2† | 10.4† | **45.8** |
| CosStruct-BI | 13 14 15 | 82.8 | 55.7 | 55.9 | **37.5** | **23.1**† | 16.2 | 45.2 |
| TreeEdit-BI | 13 16 17 | 82.5† | 59.7† | 63.1† | 17.1† | 20.0† | 15.7† | 43.0 |
| EdgeEdit-BI | 14 15 16 | **84.2** | 53.4† | 54.0† | 31.1† | 22.8† | 16.9 | 43.7 |

## 6 Limitation

**Comparison with linguistic chunks.** In Section 4.1, while we analyze subtrees inspired by chunks, which are properties of language, the analysis is limited to observing the subtree features rather than comparing them with linguistic chunks because chunks in language depend on language users. However, our primary goal is to analyze how LMs organize input tokens in the representation space at each layer; our findings provide insights into this.

**Token similarity metrics.** This study measures inter-token similarity using reciprocal L2 distance. While other metrics, e.g., cosine similarity, can be used to measure it, our objective is to analyze the organization of token representations in Euclidean space, where the magnitude carries meaningful information. Therefore, we employ the L2 distance-based metric for StructLens as a first step. Investigating other metrics can be a future direction for expanding StructLens.

**Forward constraint.** Given the left-to-right information flow of autoregressive LMs, we apply the forward constraint before constructing MSTs, which may limit the analysis of tree features, e.g., the root node. However, as we demonstrate in Section 3, this constraint preserves pairwise token similarity. Rather, this constraint makes the tree construction more efficient and yields a simple tree, making the tree causal order-respecting and easier to analyze.

**Layer pruning results.** The layer pruning results show that the structural metrics are inconsistently robust across datasets and models, limiting their applicability to model optimization. These results also indicate that there is room to improve layer pruning metrics and that structural metrics may provide insights into them.

## 7 Conclusion

Our experimental results reveal that StructLens trees exhibit a middle-layer increase in local-span connectivity under natural token order. This pattern is reduced under token shuffling, suggesting that the observed connectivity is sensitive to natural sequence organization rather than arising solely from the tree construction. Pre-training checkpoint analysis reveals the evolution of models' token representation organization in the representation space. Although, as discussed in Section 6, there are still limitations, e.g., layer pruning, our findings demonstrate that StructLens provides valuable insights into interpretability and model analysis and has the potential to expand research in this field.

## Acknowledgement

We thank the anonymous reviewers and the area chair for their valuable comments and suggestions. This work was supported by JST BOOST, Grant Number JPMJBS2423.

## References

[1] Noam Chomsky. *Syntactic Structures*. Mouton, 1957.

[2] Michael Tomasello. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, March 2005.

[3] Joan L Bybee. From usage to grammar: The mind' s response to repetition. *Language*, 82(4):711–733, December 2006.

[4] Joan Bybee. *Language, Usage and Cognition*. Cambridge University Press, 2010.

[5] Jin Hwa Lee, Thomas Jiralerspong, Lei Yu, Yoshua Bengio, and Emily Cheng. Geometric signatures of compositionality across a language model's lifetime. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5292–5320, Vienna, Austria, July 2025. Association for Computational Linguistics.

[6] nostalgebraist. Interpreting GPT: The logit lens, August 2020.

[7] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, Vienna, Austria, July 2025. Association for Computational Linguistics.

[9] Jiachen Jiang, Jinxin Zhou, and Zhihui Zhu. Tracing representation progression: Analyzing and enhancing layer-wise similarity. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Alessandro Raganato and Jörg Tiedemann. An analysis of encoder representations in transformer-based machine translation. In Tal Linzen, Grzegorz Chrupała, and Afra Alishahi, editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[11] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors, *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[12] Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. Attention can reflect syntactic structure (if you let it). In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3031–3045, Online, April 2021. Association for Computational Linguistics.

[13] Ruochen Zhang, Qinan Yu, Matianyu Zang, Carsten Eickhoff, and Ellie Pavlick. The same but different: Structural similarities and differences in multilingual language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[15] Jacob Andreas. Measuring compositionality in representation learning. In *International Conference on Learning Representations*, 2019.

[16] Xiang Lisa Li and Jason Eisner. Specializing word embeddings (for parsing) by information bottleneck. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2744–2754, Hong Kong, China, November 2019. Association for Computational Linguistics.

[17] Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D Manning. Characterizing intrinsic compositionality in transformers with tree projections. In *The Eleventh International Conference on Learning Representations*, 2023.

[18] Frederikus Hudi, Zhi Qu, Hidetaka Kamigaito, and Taro Watanabe. Disentangling pretrained representation to leverage low-resource languages in multilingual machine translation. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4978–4989, Torino, Italia, May 2024. ELRA and ICCL.

[19] Taiga Someya, Ryo Yoshida, Hitomi Yanaka, and Yohei Oseki. Derivational probing: Unveiling the layer-wise derivation of syntactic structures in neural language models. In Gemma Boleda and Michael Roth, editors, *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 93–104, Vienna, Austria, July 2025. Association for Computational Linguistics.

[20] Jason M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[21] Hiroyasu Yamada and Yuji Matsumoto. Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France, April 2003.

[22] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In Kevin Knight, Hwee Tou Ng, and Kemal Oflazer, editors, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[23] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[24] Daniel Probst and Jean-Louis Reymond. Visualization of very large high-dimensional data sets as minimum spanning trees. *Journal of Cheminformatics*, 12(1), February 2020.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Red Hook, NY, USA, December 2017. Curran Associates Inc.

[26] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[27] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A Mathematical Framework for Transformer Circuits, 2021.

[28] Hidetaka Kamigaito, Ying Zhang, Jingun Kwon, Katsuhiko Hayashi, Manabu Okumura, and Taro Watanabe. Diversity of transformer layers: One aspect of parameter scaling laws, 2025.

[29] Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[30] Karthik Viswanathan, Yuri Gardinazzi, Giada Panerai, Alberto Cazzaniga, and Matteo Biagetti. The intrinsic dimension of prompts in internal representations of large language models. *Transactions on Machine Learning Research*, 2026.

[31] Yanhong Li, Ming Li, Karen Livescu, and Jiawei Zhou. On the predictive power of representation dispersion in language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[32] Yifei Yang, Zouying Cao, and Hai Zhao. LaCo: Large language model pruning via layer collapse. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6401–6417, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[33] Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. Survey Certification, Expert Certification.

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

[35] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Analyzing feed-forward blocks in transformers through the lens of attention maps. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models, 2025.

[37] Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Lei Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[38] Emmanuel Ameisen*, Jack Lindsey*, Adam Pearce*, Wes Gurnee*, Nicholas L. Turner*, Brian Chen*, Craig Citro*, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson*. Circuit Tracing: Revealing Computational Graphs in Language Models. https://transformer-circuits.pub/2025/attribution-graphs/methods.html, 2025.

[39] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 76033–76060. Curran Associates, Inc., 2023.

[40] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[41] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019.

[42] Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. A tale of a probe and a parser. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online, July 2020. Association for Computational Linguistics.

[43] Karolina Stanczak, Edoardo Ponti, Lucas Torroba Hennigen, Ryan Cotterell, and Isabelle Augenstein. Same neurons, different languages: Probing morphosyntax in multilingual pre-trained models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1589–1598, Seattle, United States, July 2022. Association for Computational Linguistics.

[44] Jannik Brinkmann, Chris Wendler, Christian Bartelt, and Aaron Mueller. Large language models share representations of latent grammatical concepts across typologically diverse languages. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6131–6150, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[45] Michael Hanna and Aaron Mueller. Incremental sentence processing mechanisms in autoregressive transformer language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3181–3203, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[46] R. E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, March 1977.

[47] Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400, 1965.

[48] Jack Edmonds et al. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240, 1967.

[49] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna

Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024.

[50] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.

[51] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[52] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics.

[53] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[54] Eghbal A. Hosseini and Evelina Fedorenko. Large language models implicitly learn to straighten neural sentence trajectories to construct a predictive representation of natural language. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[55] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025.

[56] Cody Blakeney, Mansheej Paul, Brett W. Larsen, Sean Owen, and Jonathan Frankle. Does your data spark joy? performance gains from domain upsampling at the end of training. In *First Conference on Language Modeling*, 2024.

[57] Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James Validad Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*, 2025.

[58] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18:1245–1262, 12 1989.

[59] Benjamin Paaßen. Revisiting the tree edit distance and its backtracing: A tutorial, 2022.

[60] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019.

[61] Le Song, Alex Smola, Arthur Gretton, Karsten M. Borgwardt, and Justin Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 823–830, New York, NY, USA, 2007. Association for Computing Machinery.

[62] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[63] Christopher Wolfram and Aaron Schein. Layers at similar depths generate similar activations across LLM architectures. In *Second Conference on Language Modeling*, 2025.

[64] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Dan Roberts. The unreasonable ineffectiveness of the deeper layers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[65] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[66] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[67] Han Wu, Mingjie Zhan, Haochen Tan, Zhaohui Hou, Ding Liang, and Linqi Song. VCSUM: A versatile Chinese meeting summarization dataset. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6065–6079, Toronto, Canada, July 2023. Association for Computational Linguistics.

[68] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[69] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June 1947.

[70] Wikimedia Foundation. Wikimedia downloads.

[71] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[72] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[73] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[75] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[76] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[77] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA, 2023. Association for Computing Machinery.

[78] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[79] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, August 2007.

[80] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[81] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Information and Computation*, 82(1):93–133, 1989.

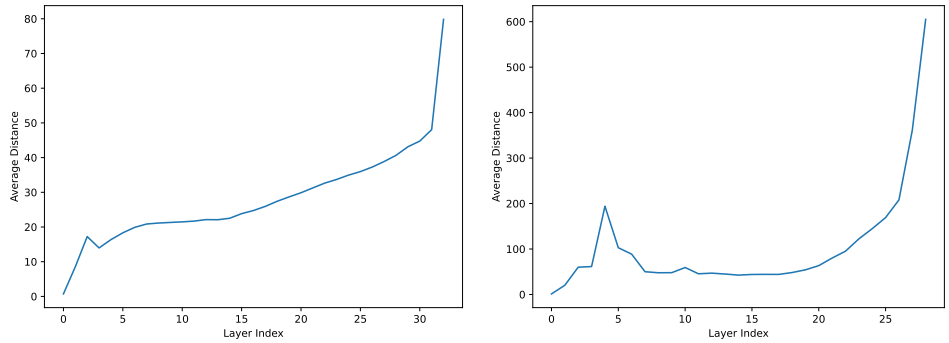
[82] Kenji Abe, Shinji Kawasoe, Tatsuya Asai, Hiroki Arimura, and Setsuo Arikawa. Optimized substructure discovery for semi-structured data. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, page 1–14, Berlin, Heidelberg, 2002. Springer-Verlag.

[83] M.J. Zaki. Efficiently mining frequent trees in a forest: algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1021–1035, 2005.

[84] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

# A Inter-Token Similarity

We compute inter-token L2 distance for Llama3.1 8B and Qwen2.5 7B using 10 samples in the Wikipedia dataset as a preliminary study. Figure 7 shows that the average in a higher layer of Qwen2.5 7B reaches approximately 600. Given this result, we convert distance into similarity by reciprocal transformation rather than exponentiation to avoid numerical underflow, as used in Equation 2.




(a) Llama3.1 8B (b) Qwen2.5 7B

Figure 7: Average L2 distance between tokens.

# B Experimental Settings (Detail)

**Datasets and Prompts.** We randomly sample 100 instances from each dataset and use the following prompt template with five-shot examples for each dataset, using the development set samples for MMLU and CMMLU. For Multinews and VCSUM, we use the prompts used in LongBench [53].

MMLU

The following are multiple-choice questions about {subject}. Respond with either A, B, C, or D as your answer.
{Question of Example1}
(A) {Choice A of Example1}
(B) {Choice B of Example1}
(C) {Choice C of Example1}
(D) {Choice D of Example1}
Answer: {Answer of Example1}
...
{Question}
(A) {Choice A}
(B) {Choice B}
(C) {Choice C}
(D) {Choice D}
Answer:

CMMLU

以下是关于（" {subject} "）的单项选择题，请直接给出正确答案的选项。
{Question of Example1}
(A) {Choice A of Example1}
(B) {Choice B of Example1}
(C) {Choice C of Example1}
(D) {Choice D of Example1}
Answer: {Answer of Example1}
...
{Question}
(A) {Choice A}
(B) {Choice B}
(C) {Choice C}
(D) {Choice D}
Answer:

Multinews

You are given several news passages. Write a one-page summary of all news.
News:
{context}
Now, write a one-page summary of all the news.
Summary:

VCSUM

下面有一段会议记录，请你阅读后，写一段总结，总结会议的内容。
会议记录：
{context}
会议总结：

**Implementations.** In this study, we use models and datasets via HuggingFace, and Tables 2a and 2b show the HuggingFace IDs of each model and dataset, respectively. Table 2c provides model revisions for each checkpoint of OLMo2 7B. To run Llama3.1 70B and Qwen2.5 72B, we quantize models to 4bit with QLoRA [71] via Transformers [72]. We use Transformers to use LMs with PyTorch [73], TensorFlow Text to build MSTs, scikit-learn [74] for spectral clustering and computing ARI, and SciPy [75] to compute correlation. To run ShortGPT [8], we employ its official implementation. Hyperparameters used in experiments are provided in Table 2d. We use a single NVIDIA GeForce RTX 3090 GPU, a single NVIDIA A100-SXM4-40GB GPU, one or two NVIDIA RTX A6000 or NVIDIA RTX 6000 Ada Generation GPUs for extracting hidden states. In the layer pruning experiments, we use lm-evaluation-harness [76] and vLLM [77] on a single NVIDIA H100 GPU or a single NVIDIA RTX Pro 6000 GPU. Code is provided in the supplement materials, and we will release the StructLens code as an installable Python library.

# C Contiguous Subtree Analysis

## C.1 Contiguous Subtrees of 3-, 4-, and 5-node

To examine whether the middle-layer signature in Section 4.1 is a specific feature of 4-node subtrees, we measure the contiguous subtree ratio and the contiguous token ratio for 3- and 5-node subtrees. Figures 8 and 9 show the layer-wise evolution of both the contiguous subtree ratio and the contiguous token ratio for 3-, 4-, and 5-node subtrees. The models exhibit the middle-layer signature across subtree sizes, suggesting that they show high contiguous token connectivity within local spans in the middle layers' representation space.

Table 2: HuggingFace ID and Hyperparameters.

(a) Models

| Model | HuggingFace ID |
|---|---|
| Llama3.1 8B | meta-llama/Llama-3.1-8B |
| Llama3.1 8B Instruct | meta-llama/Llama-3.1-8B-Instruct |
| Llama3.1 70B | meta-llama/Llama-3.1-70B |
| Qwen2.5 7B | Qwen/Qwen2.5-7B |
| Qwen2.5 7B Instruct | Qwen/Qwen2.5-7B-Instruct |
| Qwen2.5 72B | Qwen/Qwen2.5-72B |
| OLMo2 7B | allenai/OLMo-2-1124-7B |

(b) Datasets

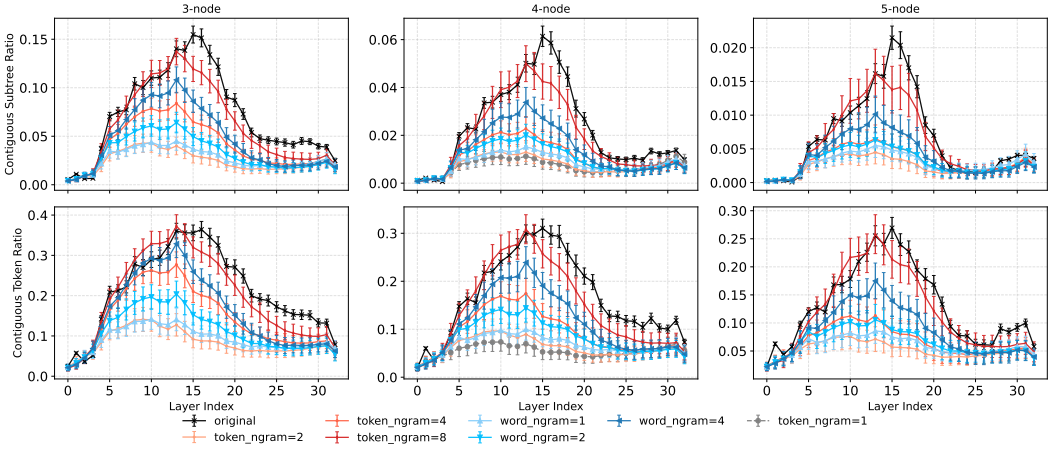
| Dataset | HuggingFace ID |
|---|---|
| MMLU | cais/mmlu |
| CMMLU | lmlmcat/cmmlu |
| Wikipedia | wikimedia/wikipedia |
| Multinews | zai-org/LongBench |
| VCSUM | zai-org/LongBench |

(c) Checkpoints

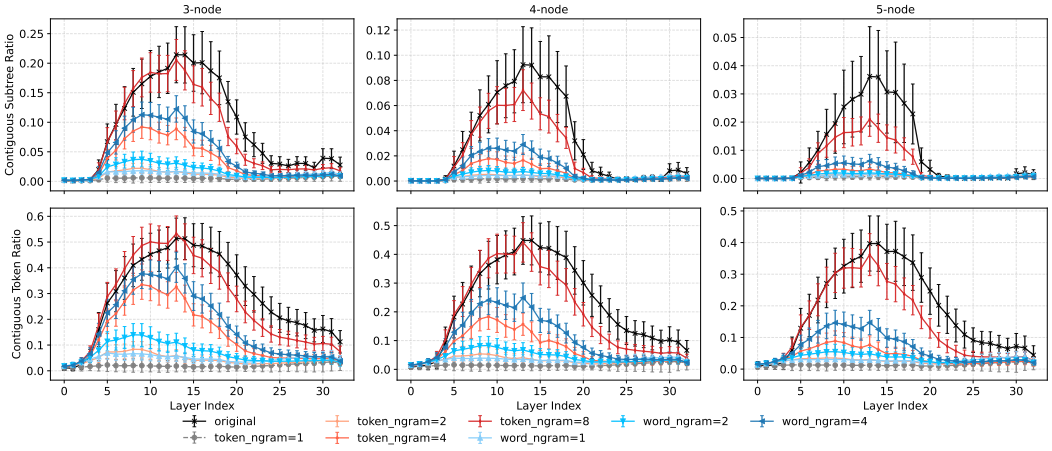
| Checkpoint | Revision (HuggingFace) |
|---|---|
| stage1-1k | stage1-step1000-tokens5B |
| stage1-10k | stage1-step10000-tokens42B |
| stage1-101k | stage1-step101000-tokens424B |
| stage1-500k | stage1-step500000-tokens2098B |
| stage1-750k | stage1-step750000-tokens3146B |
| stage1-928k | stage1-step928646-tokens3896B |
| stage2-1k | stage2-ingredient1-step1000-tokens5B |

(d) Hyperparameters

| Parameter | Value |
|---|---|
| Decoding | Greedy |
| Precision | BF16 |
| Seed | 42 |

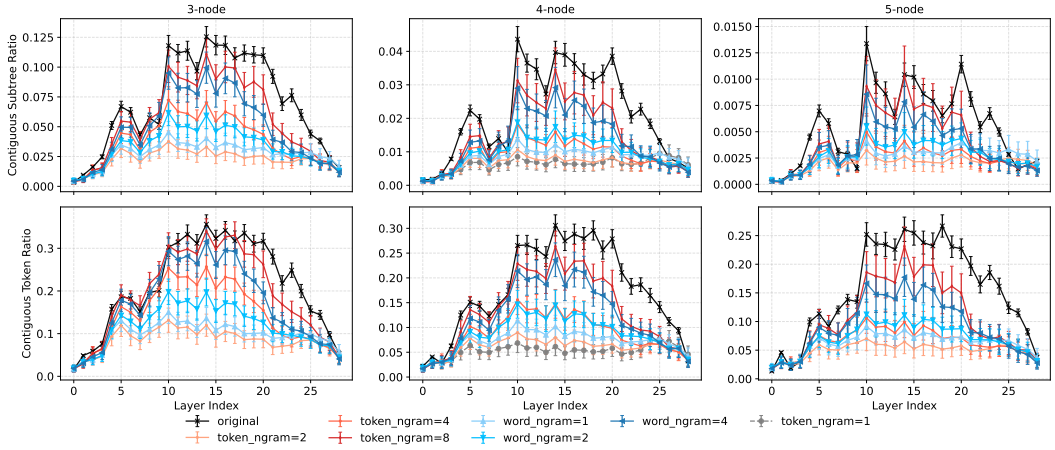



(a) MMLU

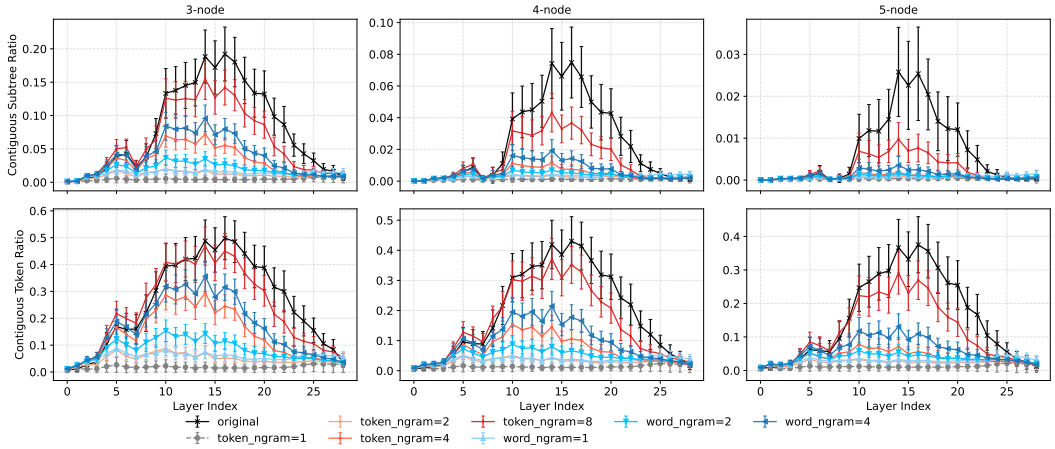



(b) Multinews

Figure 8: Contiguous subtree and token ratios for 3-, 4-, and 5-node subtrees of Llama3.1 8B.




(a) MMLU




(b) Multinews

Figure 9: Contiguous subtree and token ratios for 3-, 4-, and 5-node subtrees of Qwen2.5 7B.

## C.2 Contiguous Subtrees during Pre-training

We observe the pre-training step-wise evolution of the local-span connectivity in Section 4.2. Figure 10 shows the 5-node contiguous subtree ratio at Layer 18 and 5-shot MMLU performance across checkpoints, indicating that the contiguous subtree ratio increases sharply between stage1-500k and stage1-928k, while MMLU performance improves less. For further investigation, we shuffle input tokens and compute the contiguous subtree ratio and the contiguous token ratio as in Section 4.1. Figure 11 shows the shuffling results along with those of the original order shown in Figure 5 in Section 4.2. This figure suggests that both ratios in the middle and higher layers increase across checkpoints, indicating that the observed contiguous structures depend on the original token order rather than arising solely from implicit positional biases.

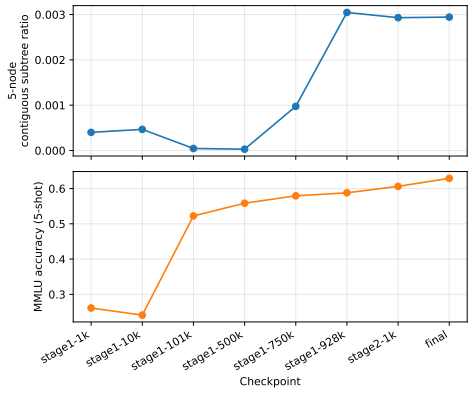



Figure 10: 5-node contiguous subtree ratio at Layer 18 and MMLU performance across checkpoints.

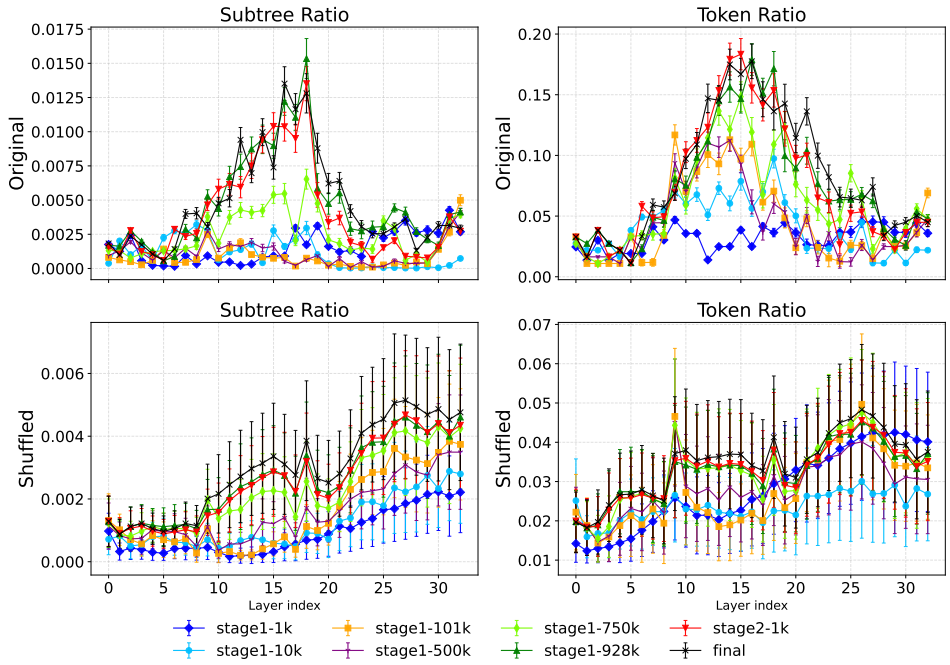
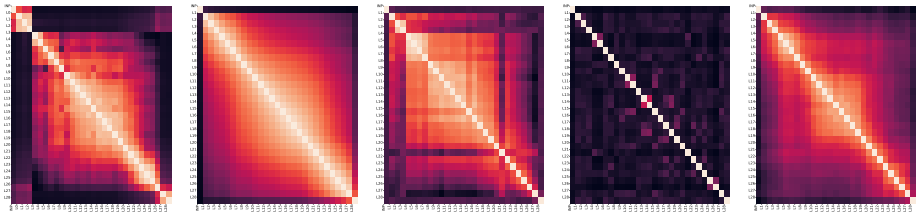



Figure 11: Visualization of the layer-wise evolution of the contiguous subtrees and tokens in them of multiple checkpoints of OLMo2 7B for MMLU. The x-axis denotes the layer index, and the y-axis denotes the contiguous subtree and token ratio.



(a) CKA (b) Cos-Base (c) Cos-Struct (d) Tree-Edit (e) Edge-Edit

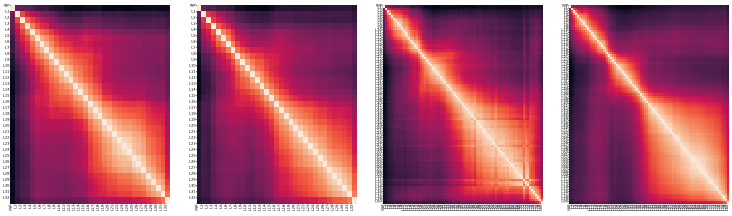
Figure 12: Inter-layer similarity samples of Qwen2.5 7B for each metric on MMLU. Bright color represents high similarity, while dark color represents low similarity.

## D Inter-Layer Similarity in Diverse Settings

**Inter-layer similarity in Qwen2.5 7B on MMLU.** We measure inter-layer similarity of CKA, Cos-Base, Cos-Struct, Tree-Edit, and Edge-Edit for Qwen2.5 7B on MMLU and investigate if it differs from that for Llama3.1 8B reported in Section 5.3. Figure 12 shows that Edge-Edit exhibits diagonal clustering patterns that we refer to as islands, as well as Llama3.1 8B, while CKA and Cos-Struct show different patterns. This suggests that Edge-Edit provides potentially insightful patterns of inter-layer similarity across models.

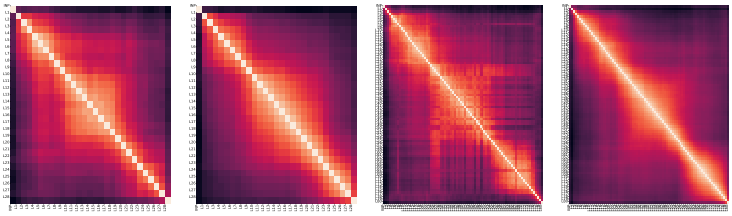
**Edge-Edit across datasets, model families, and model sizes.** To examine whether inter-layer similarity patterns by Edge-Edit shown in Section 5.3 vary across datasets, model families, and model sizes, we measure the similarity for Llama3.1 70B, Qwen2.5 7B, and Qwen2.5 72B on both MMLU and CMMLU. Figures 13 and 14 show that the similarity patterns are close within a model family, but slightly differ across model families. This result suggests that the island patterns derived by Edge-Edit are an inherent property of a model family.

**OLMo2 checkpoints.** To investigate the evolution of the island patterns during training, we measure the Edge-Edit similarity for several checkpoints, which are illustrated in Figure 15. The checkpoint



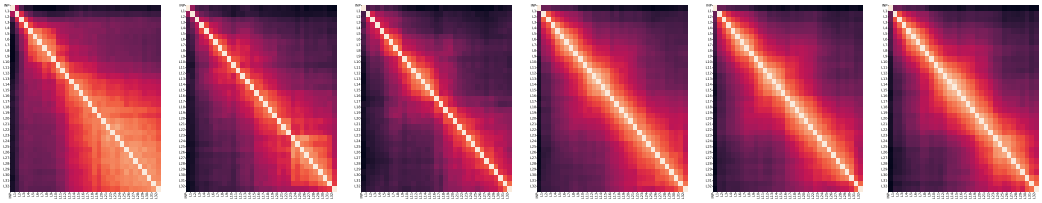
(a) 8B, MMLU (b) 8B, CMMLU (c) 70B, MMLU (d) 70B, CMMLU

Figure 13: Inter-layer similarity samples by Edge-Edit of Llama3.1 8B and Llama3.1 70B for MMLU and CMMLU. Bright color represents high similarity, while dark color represents low similarity.



(a) 7B, MMLU (b) 7B, CMMLU (c) 72B, MMLU (d) 72B, CMMLU

Figure 14: Inter-layer similarity samples by Edge-Edit of Qwen2.5 7B and Qwen2.5 72B for MMLU and CMMLU. Bright color represents high similarity, while dark color represents low similarity.



(a) stage1-1k (b) stage1-10k (c) stage1-500k (d) stage1-928k (e) stage2-1k (f) final

Figure 15: Inter-layer similarity samples of OLMo2 7B for each checkpoint on MMLU. Bright color represents high similarity, while dark color represents low similarity.

stage1-1k shows relatively large islands in the higher layers, and the model reduces their size and increases similarity between adjacent layers as training progresses. This tendency suggests that the model learns to collaborate between adjacent layers at late training steps, based on internal structures of semantic representations.

## E Layer Similarity Pattern Consistency Across Samples

We apply spectral clustering [78, 79] to partition layers into several clusters and evaluate whether the "islands" patterns are consistent across samples. For the resulting clusters, we compute the Adjusted Rand Index (ARI) [80] to measure cluster similarity between samples and employ the conductance [81] to assess the independence of each cluster.

**Clustering evaluation metrics** We formally define the conductance metric as follows. Given $l$ layers of a model, let $\mathbb{V} = \{\ell_1, \ldots, \ell_l\}$ be the set of nodes, $\mathbb{L}$ be the set of layers in a resulting cluster, and $\overline{\mathbb{L}}$ be the complement. The conductance $\varphi$ of the cluster is defined as:

$$\varphi(\mathbb{L}) = \frac{a(\mathbb{L},\, \overline{\mathbb{L}})}{\min\left(\mathrm{vol}(\mathbb{L}),\, \mathrm{vol}(\overline{\mathbb{L}})\right)}\,, \tag{12}$$

where

$$\mathrm{vol}(\mathbb{A}) = a(\mathbb{A},\, \mathbb{V})\,, \quad a(\mathbb{A}, \mathbb{B}) = \sum_{i \in \mathbb{A},\, j \in \mathbb{B}} \mathrm{score}_*(i, j)\,. \tag{13}$$

Table 3: Adjusted Rand Index (ARI) and Conductance (Cond.) on Llama 3.1 8B and Qwen2.5 7B. Bold denotes the best performance within each method.

| Method | k | Llama3.1 8B | | | | Qwen2.5 7B | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MMLU | | CMMLU | | MMLU | | CMMLU | |
| | | ARI ↑ | Cond. ↓ | ARI ↑ | Cond. ↓ | ARI ↑ | Cond. ↓ | ARI ↑ | Cond. ↓ |
| CKA | 2 | $.84_{\pm.171}$ | $.70_{\pm.046}$ | $.60_{\pm.256}$ | $\mathbf{.60_{\pm.075}}$ | $.92_{\pm.098}$ | $.86_{\pm.019}$ | $\mathbf{.88_{\pm.175}}$ | $\mathbf{.43_{\pm.027}}$ |
| | 3 | $\mathbf{1.0_{\pm.016}}$ | $\mathbf{.64_{\pm.001}}$ | $\mathbf{.67_{\pm.177}}$ | $.61_{\pm.024}$ | $.92_{\pm.077}$ | $\mathbf{.81_{\pm.017}}$ | $.53_{\pm.267}$ | $.50_{\pm.057}$ |
| | 4 | $.97_{\pm.039}$ | $.74_{\pm.001}$ | $.65_{\pm.209}$ | $.70_{\pm.017}$ | $\mathbf{.99_{\pm.060}}$ | $.87_{\pm.007}$ | $.62_{\pm.185}$ | $.57_{\pm.028}$ |
| Cos-Base | 2 | $1.0_{\pm.017}$ | $\mathbf{.48_{\pm.001}}$ | $\mathbf{.97_{\pm.051}}$ | $\mathbf{.47_{\pm.010}}$ | $.83_{\pm.220}$ | $\mathbf{.64_{\pm.046}}$ | $.82_{\pm.144}$ | $\mathbf{.53_{\pm.022}}$ |
| | 3 | $\mathbf{1.0_{\pm.000}}$ | $.63_{\pm.000}$ | $.95_{\pm.054}$ | $.64_{\pm.001}$ | $\mathbf{.99_{\pm.040}}$ | $.65_{\pm.000}$ | $\mathbf{.95_{\pm.065}}$ | $.66_{\pm.003}$ |
| | 4 | $\mathbf{1.0_{\pm.000}}$ | $.73_{\pm.000}$ | $.96_{\pm.045}$ | $.73_{\pm.000}$ | $.98_{\pm.078}$ | $.74_{\pm.000}$ | $.84_{\pm.132}$ | $.75_{\pm.001}$ |
| Cos-Struct | 2 | $\mathbf{1.0_{\pm.000}}$ | $1.0_{\pm.000}$ | $\mathbf{1.0_{\pm.000}}$ | $\mathbf{.92_{\pm.009}}$ | $\mathbf{1.0_{\pm.000}}$ | $.70_{\pm.000}$ | $.75_{\pm.257}$ | $\mathbf{.92_{\pm.044}}$ |
| | 3 | $.97_{\pm.048}$ | $\mathbf{.74_{\pm.007}}$ | $.85_{\pm.175}$ | $.96_{\pm.040}$ | $.98_{\pm.074}$ | $\mathbf{.67_{\pm.013}}$ | $.75_{\pm.249}$ | $.97_{\pm.039}$ |
| | 4 | $.75_{\pm.236}$ | $.80_{\pm.036}$ | $.97_{\pm.147}$ | $.98_{\pm.035}$ | $\mathbf{1.0_{\pm.000}}$ | $.73_{\pm.000}$ | $\mathbf{1.0_{\pm.000}}$ | $.94_{\pm.001}$ |
| Tree-Edit | 2 | $.92_{\pm.097}$ | $\mathbf{.21_{\pm.012}}$ | $.28_{\pm.297}$ | $\mathbf{.44_{\pm.090}}$ | $\mathbf{.90_{\pm.097}}$ | $\mathbf{.50_{\pm.015}}$ | $.36_{\pm.375}$ | $\mathbf{.26_{\pm.100}}$ |
| | 3 | $\mathbf{.95_{\pm.066}}$ | $.36_{\pm.013}$ | $.29_{\pm.227}$ | $.47_{\pm.051}$ | $.89_{\pm.175}$ | $.60_{\pm.014}$ | $\mathbf{.54_{\pm.211}}$ | $.34_{\pm.075}$ |
| | 4 | $.88_{\pm.115}$ | $.48_{\pm.013}$ | $\mathbf{.39_{\pm.194}}$ | $.54_{\pm.034}$ | $.87_{\pm.085}$ | $.66_{\pm.009}$ | $.45_{\pm.189}$ | $.46_{\pm.060}$ |
| Edge-Edit | 2 | $\mathbf{.99_{\pm.032}}$ | $\mathbf{.45_{\pm.014}}$ | $.48_{\pm.336}$ | $.57_{\pm.132}$ | $.56_{\pm.383}$ | $\mathbf{.59_{\pm.043}}$ | $.44_{\pm.426}$ | $\mathbf{.56_{\pm.039}}$ |
| | 3 | $.91_{\pm.068}$ | $.57_{\pm.014}$ | $\mathbf{.93_{\pm.083}}$ | $\mathbf{.55_{\pm.013}}$ | $\mathbf{.93_{\pm.064}}$ | $.60_{\pm.012}$ | $\mathbf{.87_{\pm.090}}$ | $.59_{\pm.025}$ |
| | 4 | $.86_{\pm.142}$ | $.68_{\pm.016}$ | $.80_{\pm.132}$ | $.64_{\pm.013}$ | $.79_{\pm.213}$ | $.69_{\pm.008}$ | $.65_{\pm.209}$ | $.67_{\pm.015}$ |

Lower conductance means a sharper border between clusters.

**Result** Table 3 shows that clustering is consistent across samples for a given $k$ for each metric, model, and dataset, and that $k = 2$ and $k = 3$ form sharp clusters for all metrics except Cos-Struct.

# F Frequent Subtree Mining

To find what structures are built on the islands, we perform frequent subtree mining [82, 83] on an instance in MMLU as a case study. We use FREQT[2] to run frequent subtree mining and extract ordered subtrees of eight nodes from a set of ordered trees. We extract subtrees that appeared at least twice across the collection of trees constructed for each layer, i.e., subtrees observed in a minimum of two layers. The input tokens and their indices are provided in the supplemental materials.

**Frequent subtrees in islands.** Table 4 shows that both models construct subtrees of depth eight that consist of continuous tokens in the middle and higher layers, and similar patterns appear in the later positions in the higher layers. This subtree emergence pattern suggests that models construct subtrees sequentially from left to right, whereby initially formed structural representations subsequently become obsolete in the process of residual streams. The islands are consistent with the phases observed in the intrinsic dimensionality analysis [37], which revealed that models process linguistic information (e.g., syntax and semantics) in high-intrinsic-dimensionality phases. Moreover, for MMLU, Qwen2.5 7B relates choice tokens (e.g., "A") to each other in the first few layers and does not reuse these structures in later layers.

**Frequent subtrees across non-adjacent layers.** Frequent subtree patterns also reveal the reuse of structures in non-adjacent layers. Table 5 shows such instances of frequent subtrees that did not appear in several layers. The reuse of structures between adjacent layers suggests that those layers cooperate with each other during inference, as discussed in attention heads [84], and our analysis suggests that StructLens reveals non-adjacent layer collaboration in terms of internal structures.

Our findings also show that the frequent subtree patterns are different between the two models, potentially influenced by training data, indicating that bottom-up analysis approaches are appropriate to assess the internal structure of LMs.

# G Models' behavior and Structural Transformation

We investigate the "islands" phenomenon of Edge-Edit, reported in Section 5.3, from a model's behavioral perspective at each layer, using the logit lens. Focusing on the final token outputs of the logit lens in each layer in Figure 16, Llama3.1 8B demonstrates instruction-following behavior of selecting A/B/C/D for MMLU beginning at layer 18, and similar trends are observed for Qwen2.5 7B

[2] `http://chasen.org/~taku/software/freqt/`

Table 4: Frequent subtree samples. The tree is represented as a strict S-expression. The number before "_" denotes the index of the token in the input. We replaced "(" and ")" in the input tokens with "[" and "]" to run subtree mining correctly.

(a) Llama3.1 8B, MMLU

| Subtree |
|---|
| (15_,(25_.(40_.(47_.(114_.(121_approximately(1024_approximately))))))(37_])) <br> Layers: 1, 2, 3 |
| (1_The(2_following(3_are(4_multiple(5_choice(6_questions(7_about(8_college)))))))) <br> Layers: 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 |
| (520_io(521_Is(522_chem(523_ic(524_Heart(525_Disease(530_HD)(531_])))))) <br> Layers: 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 |

(b) Qwen2.5 7B, MMLU

| Subtree |
|---|
| (35_[A(40_[B(47_[C(53_[D(142_[D(246_[D(324_[D))))))(101_[A)) <br> Layers: 0, 1, 2, 3, 4 |
| (27_side(28_effect(29_of(33_is(36_](37_muscle(49_muscle))(41_))))))) <br> Layers: 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 |
| (1013_while(1014_the(1015_heart(1016_rate(1017_[(1018_the(1019_number(1020_of)))))))) <br> Layers: 21, 22, 23, 24, 25, 26, 27, 28 |

(c) Llama3.1 8B, Multinews

| Subtree |
|---|
| (16(28_Philadelphia(324_Philadelphia(360_Philadelphia(792_Philadelphia(1961_Philadelphia (2639_Philadelphia))))))(39_Nov)) <br> Layers: 0, 2, 3 |
| (1_You(2_are(3_given(4_several(5_news(15_news(18_News(19_:)))))))) <br> Layers: 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 |
| (277_police(288_Police(1138_Police(1287_Police(1628_Police(1698_police(1699_ultimately)) (2202_police)))))) <br> Layers: 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32 |

(d) Qwen2.5 7B, Multinews

| Subtree |
|---|
| (110_But(208_For(242_They(449_The(1783_The(2183_The(2507_The))))))(1899_But)) <br> Layers: 2, 3, 4, 5, 6 |
| (391_majority(392_of(393_people(394_participating(395_in(396_this(397_movement))))(398_have)))) <br> Layers: 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 |
| (111_the(112_expected(113_police(114_eviction(115_had(116_not(117_happened(118_by)))))))) <br> Layers: 22, 23, 24, 25, 26, 27, 28 |

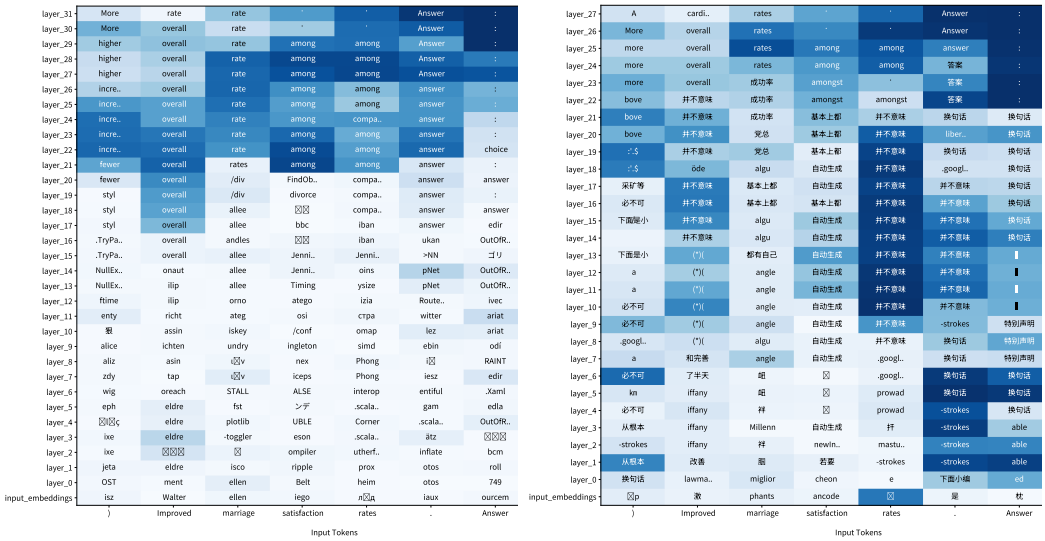
at layer 22. Table 6 examines whether this explicit transition is observed on the border of islands by spectral clustering [78, 79]. For Llama3.1 8B, layer 18 is the critical transition point, while layer 21 is the corresponding boundary for Qwen2.5 7B. These results demonstrate that structural transformations observed in StructLens in higher layers are related to the transition of token predictions.

## H Layer Pruning Results

To evaluate the layer pruning performance of each metric, i.e., CosBase-BI, CKA-BI, CosStruct-BI, TreeEdit-BI, and EdgeEdit-BI, for pre-trained models, we evaluate them on multiple-choice QA tasks that require generating only a single choice. Figure 17 shows the layer importance of each layer. Layer pruning results are shown in Table 7. These results indicate that the best metric for layer pruning is consistent within a model family.

Table 5: Frequent subtree patterns found in non-adjacent layers. This shows the top two patterns with the longest periods during which the structure was not used. The absence interval represents the maximum number of layers between when a structure observed at a layer disappears and when it reappears. For example, when a structure is observed in layers 3, 6, and 8, the absence interval is 3, corresponding to the number of layers from layer 3 to layer 6.

| Subtree |
|---|
| Llama3.1 8B |
| (72_,(86_.(414_.(471_⋯(570_⋯(788_⋯(862_⋯))))(505_.)))) <br> Layers: 1, 5, 32 <br> Absence interval: 27 |
| (14_A(36_[A(41_[B(48_[C(54_[D(143_[D(1352_[D)))(131_[C))))) <br> Layers: 1, 2, 3, 4, 5, 6, 7, 8, 29 <br> Absence interval: 21 |
| Qwen2.5 7B |
| (393_una(477_sauna(566_sauna(585_sauna(633_sauna(752_sauna(790_sauna))))(674_sauna)))) <br> Layers: 1, 2, 17, 18, 19, 20 <br> Absence interval: 15 |
| (407_by(408_short(409_-term(410_passive(411_exposure(412_to(413_extreme(414_heat)))))))) <br> Layers: 10, 11, 21, 22, 23, 24, 25, 26, 27, 28 <br> Absence interval: 10 |




(a) Llama3.1 8B (b) Qwen2.5 7B

Figure 16: Logit lens visualization on MMLU. We visualize the token predictions for each of the last eight tokens in the input. The lower rows represent predictions from the lower layers, while the upper rows show predictions from the higher layers. Color intensity represents prediction probability.

Table 6: Sample of clustering results of Edge-Edit for an instance of MMLU ($k = 3$). Layer 0 indicates the input embeddings. Cond. denotes Conductance (see Appendix E).

| | Llama3.1 8B | | Qwen2.5 7B | |
|---|---|---|---|---|
| | Layers | Cond. | Layers | Cond. |
| Cluster 1 | 0, 1, 2, 3. | .39 | 0, 1, 2, 3, 4, 5, 6, 7. | .64 |
| Cluster 2 | 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. | .76 | 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20. | .47 |
| Cluster 3 | 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32. | .44 | 21, 22, 23, 24, 25, 26, 27, 28. | .54 |

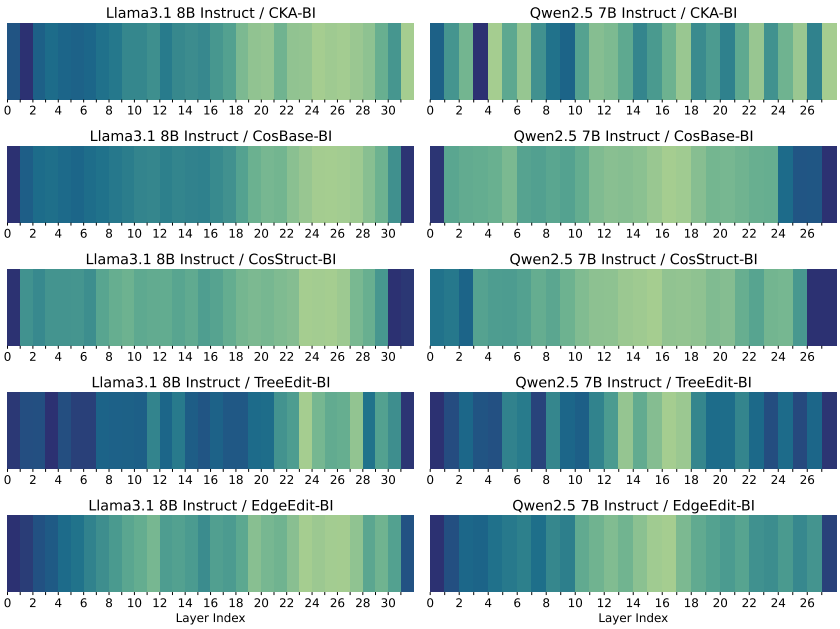



Figure 17: Visualization of layer importance. Lighter colors represent less importance, while darker colors represent greater importance.

Table 7: Pruning results of multiple-choice QA tasks. Each value denotes the accuracy or the correctness of an answer in a particular task, with higher values indicating better performance. Values denoted by † are statistically significant ($p < 0.05$) compared to CosBase-BI.

| Model | Metric | Layers removed | ARC-Easy | MMLU | CMMLU | Avg. |
|---|---|---|---|---|---|---|
| Llama-3.1-8B | Dense | - | 84.6 | 65.3 | 52.9 | 67.6 |
| | CosBase-BI | 24 25 26 27 | 76.8 | 62.0 | 49.0 | 62.6 |
| | CKA-BI | 24 26 27 31 | 72.4† | 62.3 | 49.4 | 61.4 |
| | CosStruct-BI | 23 24 25 26 | **77.8** | 64.5† | 51.2† | **64.5** |
| | TreeEdit-BI | 23 24 26 27 | 76.8 | **64.6**† | **51.9**† | 64.4 |
| | EdgeEdit-BI | 23 24 25 26 | **77.8** | 64.5† | 51.2† | **64.5** |
| Llama-3.1-8B-Instruct | Dense | - | 85.8 | 68.4 | 55.9 | 70.0 |
| | CosBase-BI | 24 25 26 27 | 78.3 | 66.8 | 54.4 | 66.5 |
| | CKA-BI | 24 25 26 31 | 75.3† | 66.9 | 54.2 | 65.5 |
| | CosStruct-BI | 23 24 25 26 | **79.3** | **67.8**† | 55.1† | **67.4** |
| | TreeEdit-BI | 23 24 26 27 | 78.1 | 66.9 | **55.8**† | 66.9 |
| | EdgeEdit-BI | 23 24 25 26 | **79.3** | **67.8**† | 55.1† | **67.4** |
| Qwen-2.5-7B | Dense | - | 86.5 | 74.3 | 82.8 | 81.2 |
| | CosBase-BI | 15 16 17 | **82.1** | 56.0 | 56.0 | 64.7 |
| | CKA-BI | 4 22 27 | 79.2† | **70.2**† | **77.7**† | **75.7** |
| | CosStruct-BI | 13 14 15 | 81.9 | 54.7† | 54.8† | 63.8 |
| | TreeEdit-BI | 12 25 26 | 71.7† | 62.9† | 72.5† | 69.0 |
| | EdgeEdit-BI | 12 15 26 | 76.3† | 60.8† | 65.5† | 67.5 |
| Qwen-2.5-7B-Instruct | Dense | - | 87.6 | 74.3 | 81.1 | 81.0 |
| | CosBase-BI | 15 16 17 | 83.7 | 56.1 | 56.8 | 65.5 |
| | CKA-BI | 4 25 27 | 80.1† | **70.0**† | **75.7**† | **75.3** |
| | CosStruct-BI | 13 14 15 | 82.8 | 55.7 | 55.9 | 64.8 |
| | TreeEdit-BI | 13 16 17 | 82.5† | 59.7† | 63.1† | 68.4 |
| | EdgeEdit-BI | 14 15 16 | **84.2** | 53.4† | 54.0† | 63.9 |

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction reflect the contribution of this paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed experimental settings to reproduce the results in Section 4, Section 5.4, and Appendix B.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We use open-access datasets in our experiments, and code is in supplemetal material. We will release it upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide detailed experimental settings to reproduce the results in Section 4, Section 5.4, and Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We show error bars in Section 4 and report the statistical significance in Section 5.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the computer resources in Appendix B.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper confirms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: This paper discusses potential impacts in Section 7

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper does not pose such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: This paper mentions existing assets properly in Sections 4 and 5.4 and Appendix B.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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