Layer-wise Positional Bias in Short-Context Language Modeling

Anonymous ACL submission

Abstract

Language models often show a preference for using information from specific positions in the input regardless of semantic relevance. While positional bias has been studied in various contexts—from attention sinks to task performance degradation in long-context settings—prior work has not established how these biases evolve across individual layers and input positions, or how they vary independent of task complexity. We introduce an attribution-based framework to analyze positional effects in short-context language modeling. Using layer conductance with a sliding-window approach, we quantify how each layer distributes importance across input positions, yielding layer-wise positional importance profiles. We find that these profiles are architecture-specific, stable across inputs, and invariant to lexical scrambling. Characterizing these profiles, we find prominent recency bias that increases with depth and subtle primacy bias that diminishes through model depth. Beyond positional structure, we also show that early layers preferentially weight content words over function words across all positions, while later layers lose this word-type differentiation.

1 Introduction

Language models rely fundamentally on word order to process linguistic structure, enabling them to capture dependencies such as grammar, compositionality, and temporal relations.

Order sensitivity arises from architectural components that enable models to learn positional information, including positional encodings such as RoPE (Su et al., 2024) and ALiBi (Press et al., 2022), causal attention masking (Vaswani et al., 2017), and position-dependent learned representations. While these mechanisms enable sequential processing, they can also introduce systematic preferences for certain input positions (Wu et al., 2025). In particular, models may assign greater influence to tokens appearing at specific positions—such as the beginning or end of the input—based on position alone rather than semantic relevance. We refer to such systematic, position-driven preferences as *positional bias*.

These positional preferences have been observed across different settings. Prior work has identified attention sinks at specific positions (Xiao et al., 2024; Han et al., 2024) and systematic position-dependent patterns in how information contributes to model predictions (Proietti et al., 2025; Wu et al., 2025). Positional bias becomes particularly evident in long-context settings, where models display the "lost in the middle" effect: when relevant information appears in the middle of lengthy inputs, models show significant performance degradation across tasks including question answering (Liu et al., 2023), retrieval (Shi et al., 2023), and summarization (Zhang et al., 2024).

However, it remains unclear how positional bias manifests across fine-grained input positions and model layers. Most existing layer-by-layer analyses are limited to long-context settings and task-level outcomes (Wu et al., 2025; Ko et al., 2020; Kongmanee, 2025). As a result, they do not establish whether positional bias reflects an intrinsic, layer-wise computational property of language models. This raises a fundamental question: if positional bias is rooted in model architecture, should it persist across layers even in short-context next-word prediction, independent of task difficulty or context length?

We address this question by characterizing positional bias at the word level (after aggregating subword tokens) for each model layer using conductance attribution in short-context next-word prediction. Unlike attention-based analyses, which primarily describe information routing, conductance quantifies how much each input token contributes to the model's prediction through intermediate model components. We apply this approach

to multiple language models and input texts using a sliding-window framework. By averaging conductance scores over words at each relative position, we obtain *layer-wise positional importance profiles* describing how layers weight positions during next-word prediction.

We investigate: (1) whether positional importance profiles are stable across inputs (a test of architectural vs. input-specific structure); (2) how *recency* and *primacy* biases evolve with layer depth; and (3) whether layers exhibit preferences for specific word types when averaged across positions.

Our contributions are fourfold:

- **Attribution framework for positional bias.** We introduce an attribution-based framework that isolates positional effects from word identity, yielding word- and layer-resolved importance scores at each position in the input.
- **Text-invariant positional importance profiles.** We show that transformer layers exhibit stable positional importance profiles in short-context next-token prediction. These profiles are highly consistent across natural texts and remain unchanged under lexical scrambling, indicating that they reflect intrinsic architectural properties rather than input-specific patterns.
- **Depth-dependent evolution of positional bias.** We characterize how primacy and recency biases evolve with depth, finding that primacy bias is weak and diminishes in deeper layers while recency bias increases monotonically with depth across models.
- **Layer specialization by word type and position.** We show that words across all positions most frequently receive their highest conductance from early layers, which also preferentially weight content words, whereas deeper layers increasingly dominate recent positions and exhibit reduced word-type sensitivity.

2 Related Work

2.1 Positional Bias in LLMs

Prior studies consistently report that language models prioritize information based on input position rather than semantic relevance. Gao et al. (2024) and Wang et al. (2025) document that LLMs struggle to utilize information from the middle or end of long contexts, exhibiting strong primacy and recency biases. This manifests as a pronounced "lost-in-the-middle" effect (Hsieh et al., 2024; Ravaut et al., 2024), where tokens at the input's beginning or end receive disproportionately high weight regardless of relevance. Veseli et al. (2025a) show that the strength of recency and primacy varies with context window size: primacy is strongest when relevant content spans $\leq 50\%$ of the window, while recency dominates as spans grow larger. These positional preferences appear as performance degradation across retrieval, summarization, and question-answering tasks (Ko et al., 2020). Notably, the magnitude of positional bias is model-specific, with some architectures exhibiting stronger primacy while others consistently favor late positions (Menschikov et al., 2025). In summary, prior work documents positional bias through task-level outcomes in long-context settings, where task difficulty and context length confound the underlying architectural bias, limiting fine-grained analysis of how individual layers allocate importance across positions.

2.2 Mechanistic Analyses of Positional Sensitivity in Transformers

Other studies probe transformers' internal computations to characterize how positional structure emerges in attention and representations. Xiao et al. (2024) and Han et al. (2024) identify *attention sinks*—specific positions (often the first or special tokens) that accumulate disproportionate attention weight across layers, independent of content. Wu et al. (2025) develop a graph-theoretic analysis of multi-layer attention and prove that causal masking inherently biases deeper layers toward earlier sequence positions. Pascual et al. (2021) perform layer-wise gradient attributions in BERT and observe that positional patterns persist across layers despite extensive contextual mixing. Broader mechanistic reviews note that positional encodings (e.g., RoPE, ALiBi) enable sequence awareness but can induce superposition and interference in internal representations (Marks et al., 2025). Thus, mechanistic work reveals the presence of positional structure but stops short of isolating position as a causal template and quantifying its layer-by-layer evolution during next-word prediction.

2.3 Layer-wise Importance in LLMs

A separate literature studies layer-wise attribution and importance in transformers, typically with-

out focusing on position. Ko et al. (2020) trace how learned answer-position bias propagates across BERT's layers. Pascual et al. (2021) perform layer-wise gradient attributions in BERT and find that raw attention scores often diverge from attribution distributions, yet persistent "fingerprints" remain in how tokens influence later layers. Hou and Castanon (2023) propose decoding hidden layers back to token space to produce layer-specific saliency maps that enhance interpretability. Other studies manipulate network structure to measure layer importance: Ikeda et al. (2025) redistribute or remove feed-forward subnetworks during pretraining and report that concentrating capacity in middle layers consistently improves downstream performance. Ferrando et al. (2022) introduce ALTI, which measures token-to-token interactions per layer and aggregates them into input attribution scores. Despite these advances, layer-wise analyses have not explicitly examined how positional features imprint on model behavior at each depth. Recently, Kongmanee (2025) begins to fill this gap by using a logit-lens approach across layers. Overall, while layer-wise methods have shown how information propagates, extracting a stable, position-dependent "template" of causal contribution across layers remains under-explored.

3 Methodology

This section describes how we compute position-resolved layer importance scores from transformer models and how these scores are aggregated to support the analyses presented in Section 4.

3.1 Attribution via Conductance

Objective. Our objective is to study how language models use information from different input positions when predicting the next word, and how this positional use of information is reflected across transformer layers. This requires a method that quantifies input contributions and relates them to layer-wise computations.

Attribution-based analysis. Attention weights are often used to study positional effects, but they do not measure causal contribution to model outputs and can be unreliable indicators of importance (Jain and Wallace, 2019; Serrano and Smith, 2019). Because our goal is to quantify how much information from each input position contributes to the prediction, we adopt an attribution-based approach.

Conductance. Conductance is an attribution method that extends Integrated Gradients by decomposing input attribution through intermediate network components (Dhamdhere et al., 2019). In its original formulation, conductance scores are summed across all input features, yielding a single importance score per layer for the entire input sequence.

Here, we adapt conductance to preserve word-level and positional structure. We compute conductance at the token level for each layer, where tokens refer to subword units produced by the model's tokenizer. These token-level scores are normalized within each individual input sequence across tokens and then aggregated by summing over subword tokens corresponding to the same surface word, yielding word-level layer importance scores. We perform analyses at the word level to reduce sensitivity to model-specific tokenization artifacts and to enable alignment with linguistic annotations such as part-of-speech (POS) tags.

Formally, let $f(x)$ denote the model's output logit for the predicted next token given input x, and let x' be a baseline input. For an input token indexed by s, the raw conductance of layer ℓ with respect to token s is defined as:

$$\mathrm{Cond}_\ell(s) = (x_s - x'_s) \int_0^1 \sum_{y \in N_\ell} \frac{\partial f\big(x' + \alpha(x - x')\big)}{\partial y} \cdot \frac{\partial y}{\partial x_s} \, d\alpha \tag{1}$$

where x_s denotes the embedding of token s, x'_s is the corresponding baseline embedding, and N_ℓ is the set of neurons in layer ℓ.

For each layer, token-level conductance scores are normalized across all input tokens:

$$\widetilde{\mathrm{Cond}}_\ell(s) = \frac{\mathrm{Cond}_\ell(s)}{\sum_{s'} \mathrm{Cond}_\ell(s')} \tag{2}$$

For a word w_i composed of subword tokens $S(i)$, the word-level conductance of layer ℓ is then defined as:

$$C(\ell, w_i) = \sum_{s \in S(i)} \widetilde{\mathrm{Cond}}_\ell(s) \tag{3}$$

These word-level conductance scores quantify how strongly information from each input word contributes to the model's prediction through each transformer layer.

3.2 Sliding-Window Attribution

To study how models assign importance to different input positions, we compute conductance

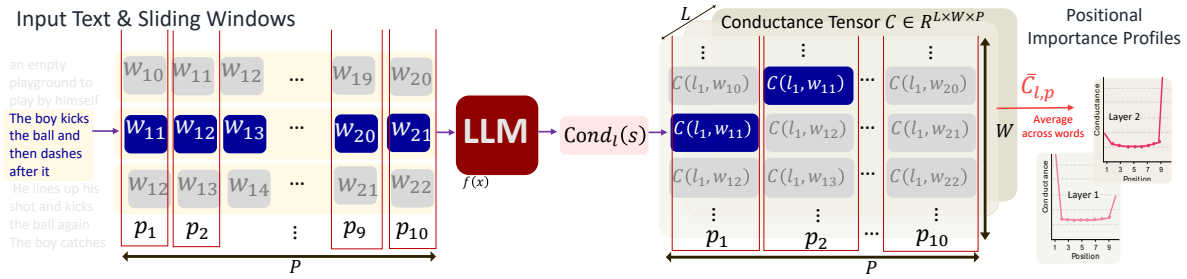

Figure 1: Conductance framework for positional bias. We move sliding windows over input text, extract layer conductance for each word-position pair from each window, store them in tensor $\mathcal{C} \in \mathbb{R}^{L \times W \times P}$, and aggregate across words to obtain positional profiles $\bar{C}_{\ell,p}$.

(our attribution method) over many inputs rather than a single sequence. However, comparing attribution scores across positions using independent sequences would be confounded by lexical and syntactic effects, as certain words or word types systematically appear in specific positions (e.g., sentence-initial tokens). To control for this effect, we use a sliding-window approach (Figure 1).

We move a window of length $P = 10$ across each text with stride one (Figure 1, left) and compute layer conductance for every window using the Captum library (Kokhlikyan et al., 2023; Miglani et al., 2023). We choose $P = 10$ to enable fine-grained positional analysis in short-context settings while ensuring sufficient context for meaningful next-token prediction. Results with $P = 50$ for two stories are provided in Appendix E, demonstrating consistency of positional patterns across window lengths.

We remove boundary words from the analysis. After this removal, each relative position contains the same set of words. This construction allows us to compare conductance scores across positions while holding word identity constant, isolating positional effects from lexical content.

For each word w and layer ℓ, we collect conductance scores by relative position:

$$\mathbf{c}_\ell(w) = \big(c_\ell(w,1), c_\ell(w,2), \ldots, c_\ell(w,P)\big),$$

where $c_\ell(w,p)$ denotes the conductance assigned by layer ℓ to word w at relative position p within a window. Collecting these vectors across all words and layers forms a three-dimensional tensor $\mathcal{C} \in \mathbb{R}^{L \times W \times P}$, where L is the number of layers, W is the number of words, and P is the window length (Figure 1, middle; the figure uses subscripted notation such as l_1, w_{11} for specific indices). This representation preserves both word identity and relative position, and forms the foundation of all subsequent analyses.

3.3 Positional Importance Profiles

To analyze how layers distribute importance across input positions, we construct positional importance profiles by averaging word-level conductance scores across words (Figure 1, right):

$$\bar{C}_{\ell,p} = \frac{1}{W} \sum_{w=1}^{W} C_{\ell,w,p}. \tag{4}$$

For each layer ℓ, the vector $\bar{C}_\ell = (\bar{C}_{\ell,1}, \ldots, \bar{C}_{\ell,P})$ summarizes how importance is allocated across relative input positions.

Between-input consistency. To verify that positional profiles are architectural rather than input-specific, we compute the mean pairwise Pearson correlation between the layer-wise positional profiles obtained from the natural texts and the randomized word-shuffled baseline.

Primacy and recency metrics. To summarize positional importance profiles using scalar measures, we define primacy and recency fractions. Let $\mathcal{P}$ and $\mathcal{R}$ denote the sets of early (primacy) and late (recency) relative positions, respectively, defined as the first and last 20% of positions within the window. For each layer ℓ, we define:

$$\text{PrimFrac}_\ell = \frac{\sum_{p \in \mathcal{P}} \bar{C}_{\ell,p}}{\sum_{p=1}^{P} \bar{C}_{\ell,p}} \qquad \text{RecFrac}_\ell = \frac{\sum_{p \in \mathcal{R}} \bar{C}_{\ell,p}}{\sum_{p=1}^{P} \bar{C}_{\ell,p}} \tag{5}$$

These measures quantify the proportion of a layer's total importance assigned to early and late input positions, respectively.

3.4 Word-level aggregation

Beyond positional structure, we analyze how layers weight different word types by averaging conductance across all positions. For each layer ℓ and word w, the position-averaged conductance is:

$$\tilde{C}_{\ell,w} = \frac{1}{P} \sum_{p=1}^{P} C_{\ell,w,p}. \tag{6}$$

To characterize word-type preferences, we group words by POS using spaCy (Honnibal et al., 2020) and compute mean importance per POS category. We distinguish between content words (nouns, proper nouns, verbs, adjectives, adverbs) and function words (pronouns, auxiliaries, determiners, prepositions, particles). For word-level analyses (Sections 4.3 and 4.4), we pool words across texts after position averaging. When grouping by POS category, we exclude the scrambled text to ensure that all texts contribute the same set of words for each category.

3.5 Word-level layer dominance

We define layer dominance by identifying, for each word and position, the layer assigning the maximum conductance score:

$$\ell^*(w, p) = \arg\max_{\ell} C_{\ell,w,p}. \quad (7)$$

We also apply layer dominance after aggregating conductance scores over positions, yielding a dominant layer for each word.

4 Experimental Setup and Results

Datasets We analyze eight texts spanning three genres: two narrative stories (*Pie Man* and *Shapes*) from the Narratives dataset (Nastase et al., 2021), three Wikipedia articles from Wikipedia Monthly (Kamali and Labs, 2025), and three scientific abstracts from SciDocs (Thakur et al., 2021) (310-957 words). We also include a scrambled version of *Pie Man* as a control. See Appendix A for details.

Models We analyze four pretrained decoder-only transformers: GPT-2 (Radford et al., 2019) (124M, 12 layers), GPT-Neo-2.7B (Black et al., 2021) (32 layers), Phi-2 (Gunasekar et al., 2023) (2.7B, 32 layers), and Llama-3-8B (Meta AI, 2024) (32 layers). For the P=50 robustness analysis (Appendix E), we use Llama-3.2 (Meta, 2024) (1B, 16 layers), instead of Llama-3-8B due to computational constraints. These models vary in scale (124M to 8B parameters), depth (12 to 32 layers), and training data (WebText, The Pile, curated synthetic data, large-scale web corpora), enabling assessment of positional bias patterns across architectures. All models use learned positional embeddings and causal masking. We use pretrained checkpoints without fine-tuning.

4.1 Layer-wise Importance Profiles Across Positions

We first examine how importance is distributed across relative input positions at different depths of the model. Figure 2 shows layer-wise positional importance profiles for several language models on the *Pie Man* story, computed using normalized layer conductance and averaged over words (results for other texts, scrambled version, and layer-position heatmaps (mean and variance) are provided in Appendix B and Appendix C).

Progressive Recency Across models, positional importance increases monotonically toward more recent input positions, with all layers assigning higher importance to tokens closer to the prediction point. This effect strengthens with depth: later layers increasingly concentrate importance on the most recent positions, and the final layer assigns all importance to the immediately preceding token. This reflects how information flows through autoregressive transformers: as later positions accumulate context from earlier positions through successive layers of attention, deeper layers can rely primarily on recent representations that already encode the necessary information from earlier tokens.

Primacy and Anchoring In addition to the dominant recency trend, GPT-2, GPT-Neo, and Llama-3 exhibit a persistent elevation of importance at the first or second position. This pattern aligns with the "Attention Sink" hypothesis (Xiao et al., 2024), suggesting that these architectures utilize initial tokens as computational anchors to stabilize internal representations. Notably, Phi-2 deviates from this pattern, showing elevated importance at initial positions only in the final few layers, suggesting architectural variation in how models process early-context information.

Mid-Window Convergence Despite architectural differences, positional importance profiles converge near the middle of the input window (approximately position 6), particularly in intermediate layers. At this position, importance values across models and layers become nearly indistinguishable, suggesting reduced sensitivity to architectural differences in the middle of the context window.

These results indicate that layers differentially weight positional information as representations are transformed through the network. In the following sections, we quantify these positional patterns,

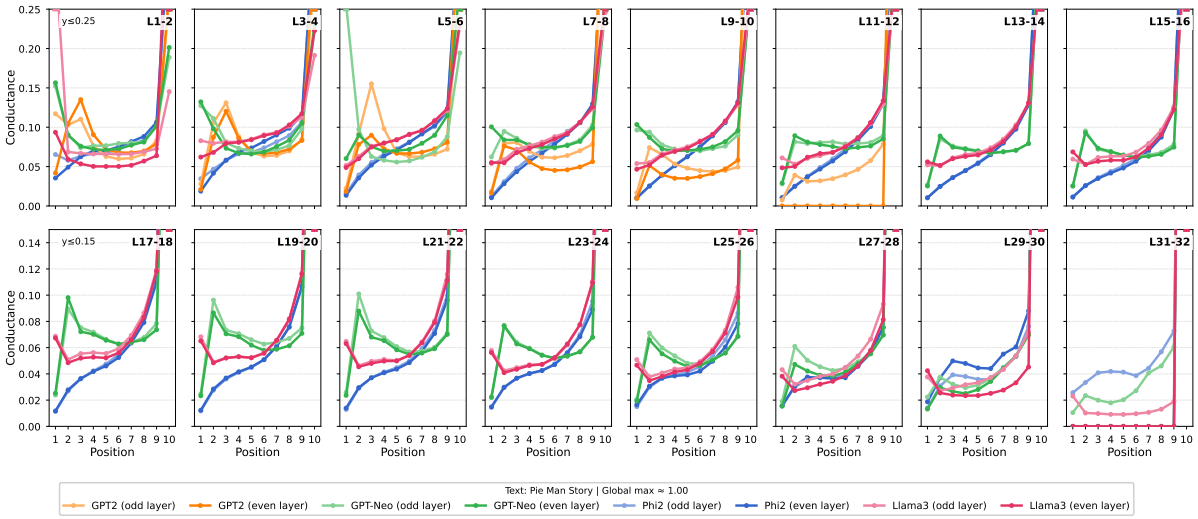

Figure 2: Layer-wise positional importance profiles across models for the *Pie Man* story. For each model and layer, curves show conductance scores averaged over words as a function of input position. Each subplot shows a consecutive layer pair (e.g., L1-2), with odd layers in lighter shades and even layers in darker shades for visual clarity. Y-axis ranges are adjusted per row (global max = 1.00). All layers exhibit a pronounced peak at recent positions and a secondary peak at primacy positions. See Appendix B for results on other texts and the scrambled version, and Appendix C for layer–position heatmaps (mean and variance).

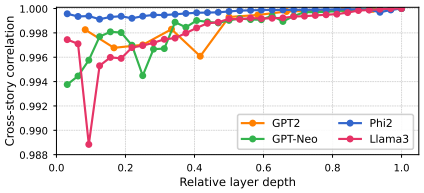

Figure 3: Cross-story consistency of positional importance profiles. Mean pairwise Pearson correlation between profiles across inputs for each layer. Correlations exceed 0.99 across all models, indicating text-invariant positional profiles.

assess their consistency across stories, and compare the strength of primacy and recency effects across models.

4.2 Positional Bias Beyond Text Content

To test whether positional profiles depend on semantic content, we compare them across eight natural texts and a scrambled control where word order is randomized.

Mean pairwise correlations exceed $r > 0.99$ for every layer across all models (Figure A22), including comparisons between natural and scrambled text. The persistence of these profiles, even when semantic and syntactic dependencies are destroyed, demonstrates that the allocation of importance is not dictated by the location of meaningful words, but by a stable architectural template.

We therefore characterize this behavior as an intrinsic positional bias: a systematic weighting of information driven primarily by relative position

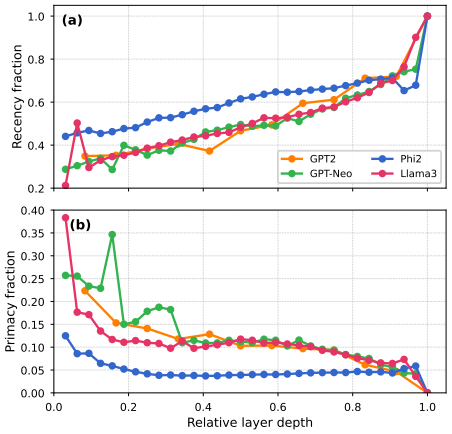

Figure 4: Evolution of positional bias across layers. (a) Recency fraction. (b) Primacy fraction. Patterns are averaged across all texts; recency increases monotonically with depth across all models.

rather than semantic content.

Layer Specialization via Positional Metrics We characterize the evolution of positional bias across models using the Recency and Primacy metrics defined in Section 3.3. Recency bias increases monotonically with depth across all models (Figure 4a). While early layers distribute causal mass broadly (recency $\approx$ 0.3–0.4), later layers progressively concentrate on the immediate context, with the final layer allocating nearly 100% of its contribution to the last token.

In contrast, primacy bias follows an inverse trend, peaking in early layers before stabilizing at lower levels in middle and late layers (Figure 4b).

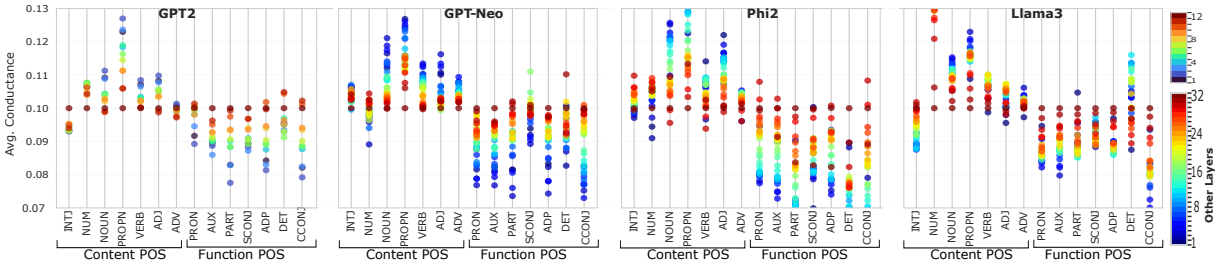

Figure 5: Position-averaged layer importance by part-of-speech (POS) category. Layer conductance is computed at the word level and averaged over positions before aggregation within POS groups.

Most layers assign less than 15% of their importance to early positions. However, the magnitude varies by architecture: Llama-3 and GPT-Neo show strong initial anchoring (primacy > 0.25), whereas Phi-2 exhibits consistently low primacy, and GPT-Neo displays a distinct oscillation in the early layers.

These patterns align with recent attribution-based findings (Proietti et al., 2025), but we extend them by establishing text-invariance and characterizing layer-wise evolution. Furthermore, while Veseli et al. (2025b) suggest that recency dominates as inputs approach the context window limit, we show that this dominance is the default operating mode of the architecture, evident even in short, unconstrained contexts. This suggests that positional bias reflects a fundamental architectural tendency, rather than arising solely as a response to challenging or long-context inputs.

While the general trend is consistent across models, the observed divergences may reflect a combination of architectural and training-related factors. Architectural choices—such as positional encoding schemes or attention mechanisms that emphasize boundary tokens—have been shown to shape positional sensitivity in transformers (Su et al., 2024; Wu et al., 2025). In addition, training data distributions may imprint systematic patterns of human communication onto model weights (Itzhak et al., 2024; Raimondi and Gabbrielli, 2025). For example, GPT-Neo's oscillatory primacy aligns with its alternating local/global attention layers (Black et al., 2021), while Phi-2's reduced primacy may reflect its curated training data (Gunasekar et al., 2023).

4.3 Word-Level Layer Importance After Positional Averaging

We previously showed that each model layer exhibits a positional importance profile, describing how its contribution to next-token prediction varies as a function of word position, largely independent of semantic content. Here, we ask a complementary question: do layers consistently allocate more importance to certain types of words, regardless of where those words appear in the input?

Using position-averaged conductance scores grouped by POS (Section 3.4), we examine layer-by-POS importance profiles (Figure 5). Patterns are nearly identical across all models.

Because the final layer concentrates all importance on the single position immediately preceding the prediction, averaging across positions yields the same mean importance for all words and POS categories in this layer ($1/P$, where P is the context window length). This value provides a reference point for interpreting deviations observed in earlier layers.

Across POS categories, early layers (blue) show substantial deviations from this reference value, whereas deeper layers (red) increasingly converge toward it. In particular, content words (e.g., nouns and verbs) receive consistently higher average importance than the reference value across multiple early and middle layers. This indicates that early layers are more sensitive to word identity and lexical content.

In contrast, function words (e.g., determiners and prepositions) receive importance values at or below the reference level across layers, indicating that they are generally less influential for prediction once positional effects are removed. As depth increases, differences between POS categories diminish, suggesting reduced word-type differentiation in later layers.

Together, these results show that word-type sensitivity is primarily a property of early processing stages. After positional effects are marginalized, early layers continue to differentiate between content and function words, while deeper layers converge toward a uniform allocation of importance in which word identity plays a limited role. This pattern indicates that positional structure increasingly dominates layer behavior with depth.

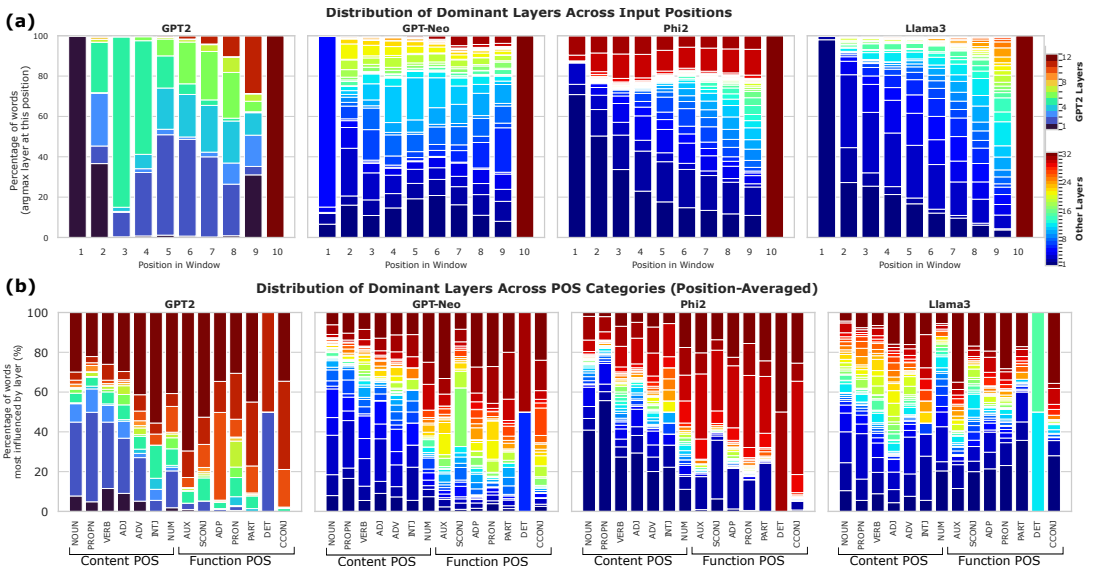

Figure 6: Distribution of dominant layers (highest conductance per word). (a) By position: percentage of words at each position dominated by each layer. (b) By part-of-speech (POS) (position-averaged): percentage of words in each POS category dominated by each layer. Early layers dominate the largest proportion of words at all positions. Later layers dominate only a small proportion, which increases toward the prediction point. Content words are dominated by early layers; function words by later layers.

4.4 Word-Level Layer Dominance

We further analyze positional effects at the level of individual words by identifying the model layer that assigns the highest conductance score to each word. We refer to this as the word's dominant layer.

Layer Dominance by Position We first computed the dominant layer for each word–position pair. Figure 6a shows the percentage of words at each position dominated by each layer, pooling words across natural texts and scrambled text. The dominant layers shift continuously with input position.

Across all models, early layers dominate the largest proportion of words at all positions. Later layers dominate only a small proportion, which increases toward the prediction point. This suggests that early layers maintain a universal information-gathering role across all input positions.

When we examined each text separately, the distribution of dominant layers at each position remained consistent (Appendix Figures D). This consistency complements but differs from Section 4.2 in an important way. Section 4.2 demonstrated that each layer's conductance profile across positions is stable (e.g., Llama-3's Layer 1 consistently shows high importance at Position 1 across all inputs). Here, we show that at each position, the relative dominance ranking of layers is stable (e.g., at Position 1, Llama-3's early layers consistently dominate). Together, these findings confirm that positional bias reflects architectural properties rather than semantic content.

Layer Dominance by POS To examine word-type effects, we averaged conductance scores across positions (Section 3.4) and identified the dominant layer for each word after this aggregation. We then grouped words by POS category (Figure 6b). Content words (e.g., nouns and verbs) were more frequently dominated by earlier layers, whereas function words (e.g., determiners and prepositions) were more frequently dominated by later layers in GPT-2, GPT-Neo, and Phi-2, consistent with prior findings in GPT-2 (Rahimi et al., 2025). This effect was weaker in Llama-3, revealing architectural variation. This aligns with our position-averaged word-level analysis (Section 4.3), where content words showed higher early-layer importance and function words showed higher late-layer importance.

4.5 Coclusion

We show that LLM layers exhibit stable positional importance profiles in short-context next-word prediction, indicating a form of positional bias that is intrinsic to model architecture. We also observe layer-specific word-level effects: early layers assign the highest conductance to most words across all positions and preferentially weight content words, whereas later layers lose word-type sensitivity and increasingly concentrate importance on recent positions.

5 Limitations

This study focuses on positional importance in short-context next-word prediction and is subject to several limitations.

First, our analyses are restricted to relatively short contexts. While this setting allows precise, token-level characterization of positional effects, the observed positional importance profiles may differ in longer contexts or in tasks that require explicit reasoning, retrieval, or discourse-level planning. Extending the analysis to long-context regimes and tasks beyond next-word prediction remains an important direction for future work.

Second, we examine a limited set of autoregressive transformer models. Although the consistency of the observed positional patterns across models suggests the effect is not model-specific, our findings may not generalize to architectures with substantially different inductive biases, such as encoder–decoder models, architectures with alternative positional encoding schemes, or models with non-standard attention mechanisms.

Third, our analysis relies on layer conductance as a proxy for layer importance. While conductance provides a principled and path-integrated attribution method, it remains an approximation of internal computation and does not establish causal necessity or identify the specific architectural components that give rise to these patterns. Different attribution methods may emphasize complementary aspects of model behavior, and our conclusions should be interpreted in the context of this methodological choice.

Finally, word-type analyses are based on part-of-speech categories and averaging procedures. Although these analyses reveal systematic layer-specific tendencies, they do not capture finer-grained syntactic distinctions (e.g., argument structure, dependency relations) or semantic distinctions (e.g., lexical semantic classes), nor do they account for interactions between word type and broader discourse structure.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *arXiv preprint arXiv:2104.04473*.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2019. How important is a neuron? In *International Conference on Learning Representations (ICLR)*.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into LLM long-context failures: When transformers know but don't tell. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA. Association for Computational Linguistics.

Suriya Gunasekar, Yi Zhang, Raviraj Das, and 1 others. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024. Lm-infinite: Simple on-the-fly length generalization for large language models. *arXiv preprint arXiv:2308.16137*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. *Zenodo*.

Elizabeth M. Hou and Gregory Castanon. 2023. Decoding layer saliency in language transformers. *Preprint*, arXiv:2308.05219.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand. Association for Computational Linguistics.

Wataru Ikeda, Kazuki Yano, Ryosuke Takahashi, Jaesung Lee, Keigo Shibata, and Jun Suzuki. 2025. Layerwise importance analysis of feed-forward networks in transformer-based language models. *Preprint*, arXiv:2508.17734.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. 2024. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *Transactions of the Association for Computational Linguistics*, 12:771–785.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of NAACL-HLT*.

Omar Kamali and Omneity Labs. 2025. Wikipedia monthly.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, and Edward Wang. 2023. Captum: A unified and generic model interpretability library for pytorch. https://github.com/pytorch/captum.

Jaturong Kongmanee. 2025. Unraveling token prediction refinement and identifying essential layers in language models. *Preprint*, arXiv:2501.15054.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. In *Transactions of the Association for Computational Linguistics*, volume 12, pages 157–173.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Preprint*, arXiv:2403.19647.

Mikhail Menschikov, Alexander Kharitonov, Maiia Kotyga, Vadim Porvatov, Anna Zhukovskaya, David Kagramanyan, Egor Shvetsov, and Evgeny Burnaev. 2025. Beyond early-token bias: Model-specific and language-specific position effects in multilingual llms. *Preprint*, arXiv:2505.16134.

Meta. 2024. Llama 3.2 1b. https://huggingface.co/meta-llama/Llama-3.2-1B. Accessed: January 2026.

Meta AI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Vivek Miglani, Oliver Aobo Yang, Aram H. Markosyan, Diego Garcia-Olano, and Narine Kokhlikyan. 2023. Using captum to explain generative language models.

Samuel A Nastase, Yun-Fei Liu, Harry Hillman, Asieh Zadbood, Lilla Hasenfratz, Nava Keshavarzian, Jiahe Chen, Christopher J Honey, Yaara Yeshurun, Mor Regev, Mai Nguyen, Catie HC Chang, Christopher Baldassano, Olga Lositsky, Erez Simony, Mark A Chow, Yida C Leong, Peter P Brooks, Elizabeth Micciche, and 6 others. 2021. The "narratives" fmri dataset for evaluating models of naturalistic language comprehension. *Scientific Data*, 8(1):1–13.

Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2021. Telling BERT's full story: from local attention to global aggregation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics.

Ofir Press, Noah A Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Michela Proietti, Roberto Capobianco, and Mariya Toneva. 2025. Fine-grained analysis of brain-llm alignment through input attribution. *Preprint*, arXiv:2510.12355.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Maryam Rahimi, Yadollah Yaghoobzadeh, and Mohammad Reza Daliri. 2025. Explanations of large language models explain language representations in the brain. *arXiv preprint arXiv:2502.14671*.

Bianca Raimondi and Maurizio Gabbrielli. 2025. Exploiting primacy effect to improve large language models. *arXiv preprint arXiv:2507.13949*. Accepted to RANLP 2025.

Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand. Association for Computational Linguistics.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of ACL*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *arXiv preprint arXiv:2302.00093*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. 2025a. Positional biases shift as inputs approach context window limits. *Preprint*, arXiv:2508.07479.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. 2025b. Positional biases shift as inputs approach context window limits. In *First Conference on Language Modeling*.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. 2025. Eliminating position bias of language models: A mechanistic approach. In *The Thirteenth International Conference on Learning Representations*.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. 2025. On the emergence of position bias in transformers. *arXiv preprint arXiv:2502.01951*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Weijie Zhang, Hanfeng Feng, Jiaqi Zhang, Zhe Chen, Liang Huang, and Yue Zhang. 2024. Benchmarking long-context language models with multiple needles in a haystack. *arXiv preprint arXiv:2406.13471*.

A Dataset Details

We analyzed eight natural texts across three genres, plus one scrambled control text.

Narrative Stories. We used two stories from the Narratives dataset (Nastase et al., 2021): *Pie Man* (957 words) and *Shapes* (910 words). This dataset contains naturalistic spoken narratives transcribed to text. We used the original word order without additional preprocessing.

Encyclopedic Articles. Three articles randomly selected from the English Wikipedia Monthly dataset (Kamali and Labs, 2025): *Alabama*, *Albedo*, *Anarchism*, with 735, 310, and 703 words respectively after cleaning.

Scientific Abstracts. Three abstracts randomly selected from the SciDocs corpus (Thakur et al., 2021): Paper titles *The importance of drawing in the mechanical design process*, *Learning Agents for Uncertain Environments (Extended Abstract)*, and *The Intersecting Roles of Consumer and Producer*, with 794, 739, and 793 words respectively.

Scrambled Control. We created a scrambled version of the *Pie Man* story by uniformly shuffling its word sequence. This control tests whether positional patterns persist independent of meaningful content.

B Positional Importance Profiles: Additional Texts

B.1 Narrative Stories

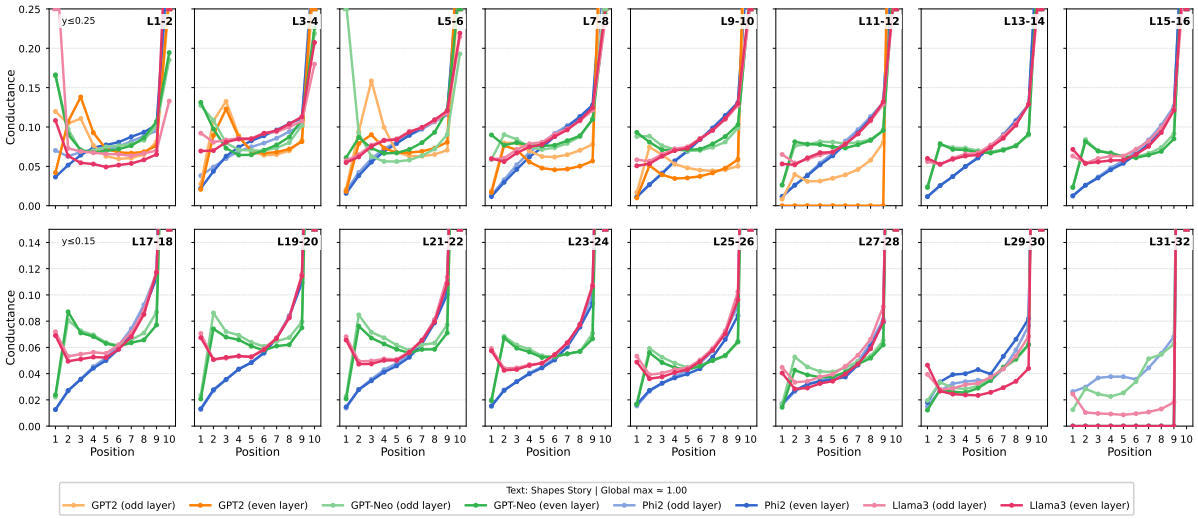

Figure A1: **Shapes:** Layer-wise positional importance profiles. Conductance scores averaged over words as a function of position. Patterns are similar to those observed in Figure 2.

B.2 Scrambled Control

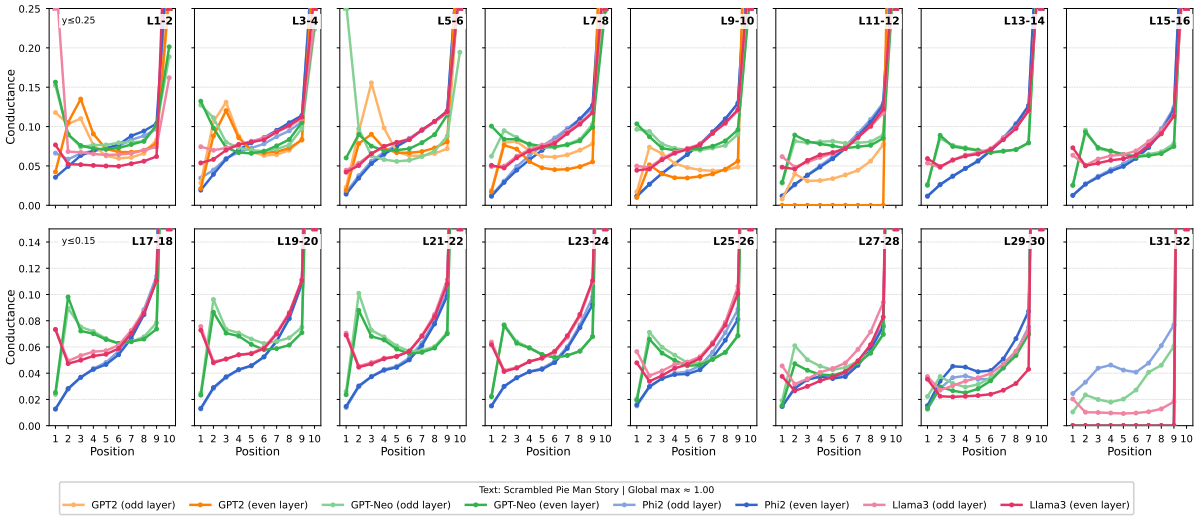

Figure A2: **Scrambled Pie Man:** Layer-wise positional importance profiles. Despite scrambling, similar recency and primacy peaks are observed.

B.3 Wikipedia Articles

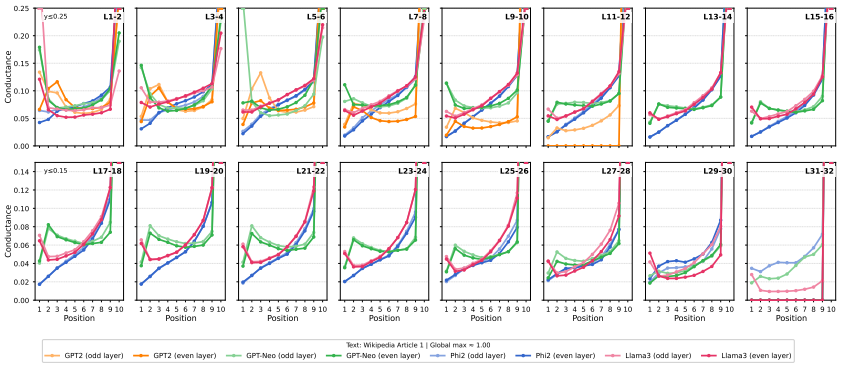

Figure A3: **Wikipedia Article 1:** Layer-wise positional importance profiles.

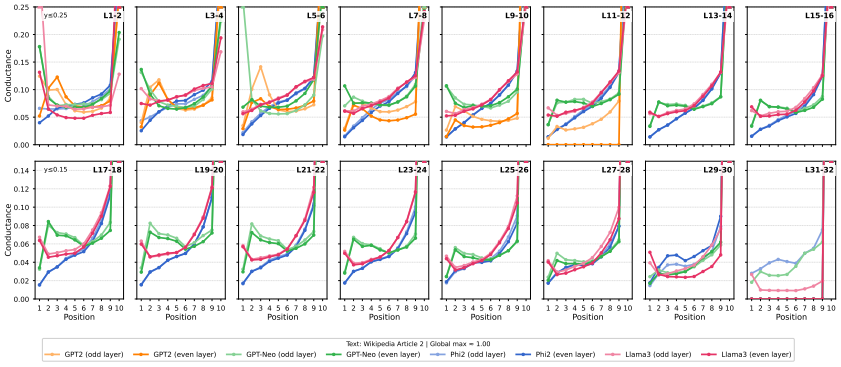

Figure A4: **Wikipedia Article 2:** Layer-wise positional importance profiles.

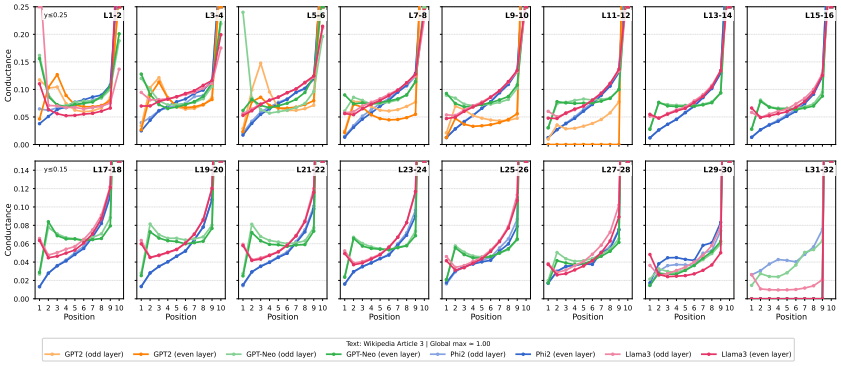

Figure A5: **Wikipedia Article 3:** Layer-wise positional importance profiles.

B.4 Scientific Abstracts

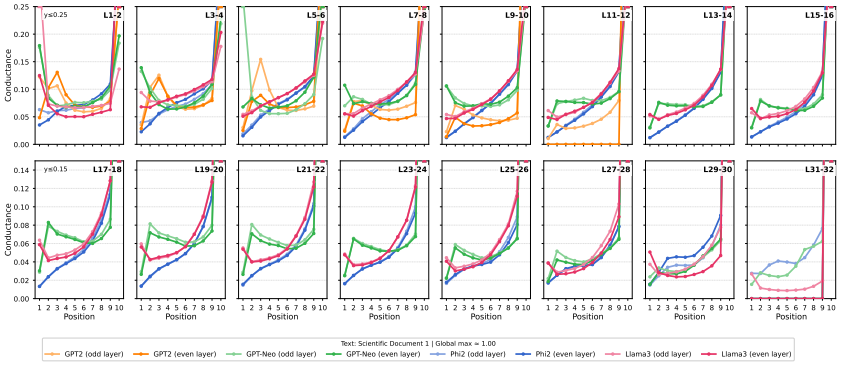

Figure A6: **Scientific Document 1:** Layer-wise positional importance profiles.

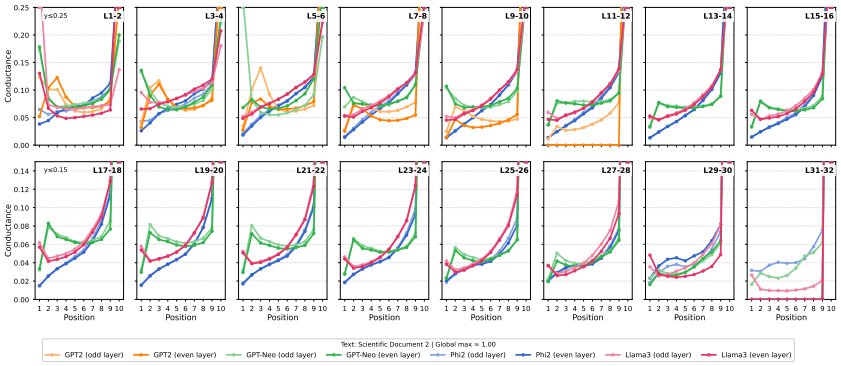

Figure A7: **Scientific Document 2:** Layer-wise positional importance profiles.

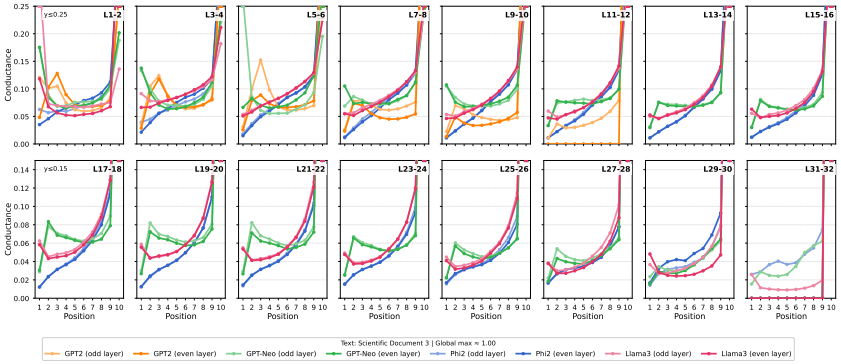

Figure A8: **Scientific Document 3:** Layer-wise positional importance profiles.

C Layer-Position Heatmaps

C.1 Mean Conductance Heatmaps

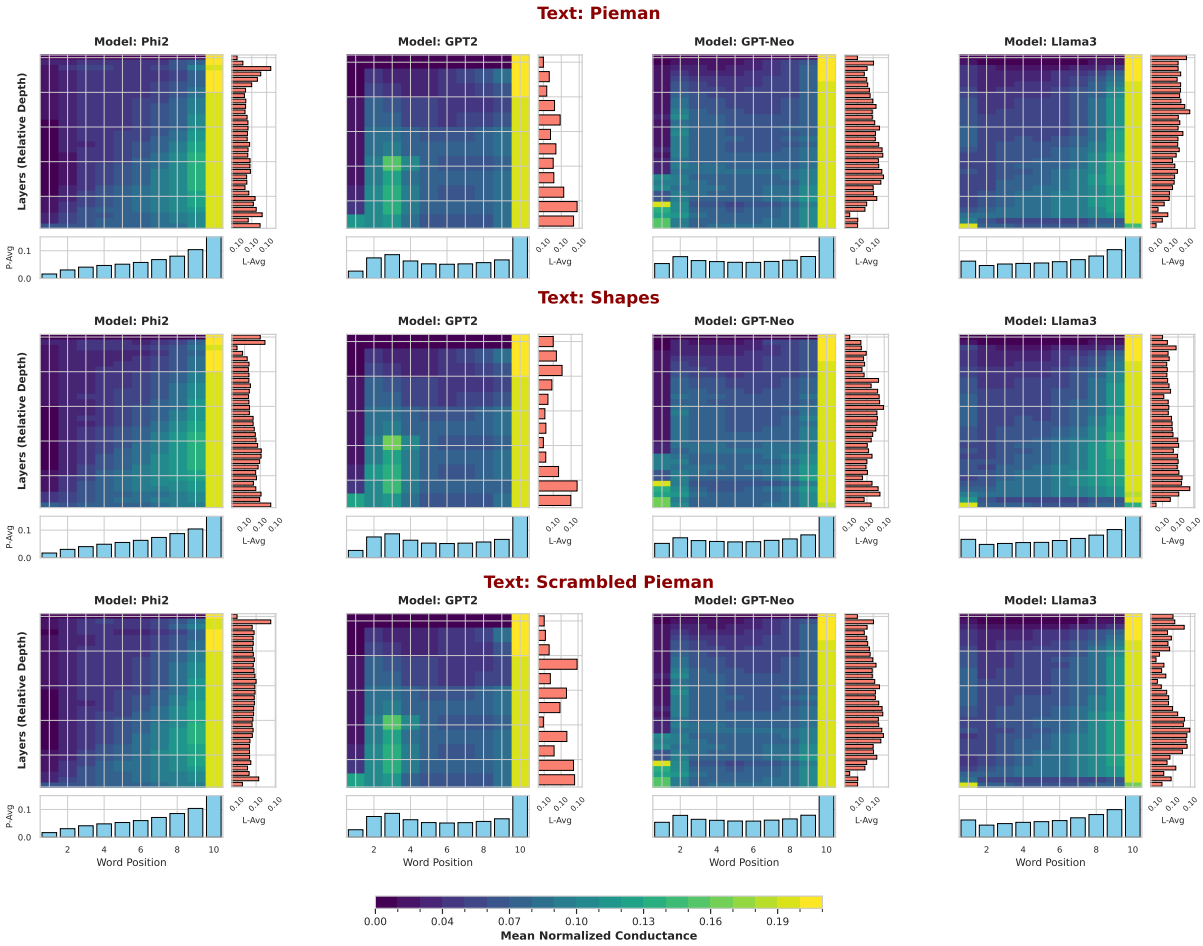

Figure A9: Layer-position heatmaps showing mean conductance. Mean conductance scores across layers (rows) and positions (columns) for each model and story. Heatmaps facilitate direct comparison of layer importance across positions.

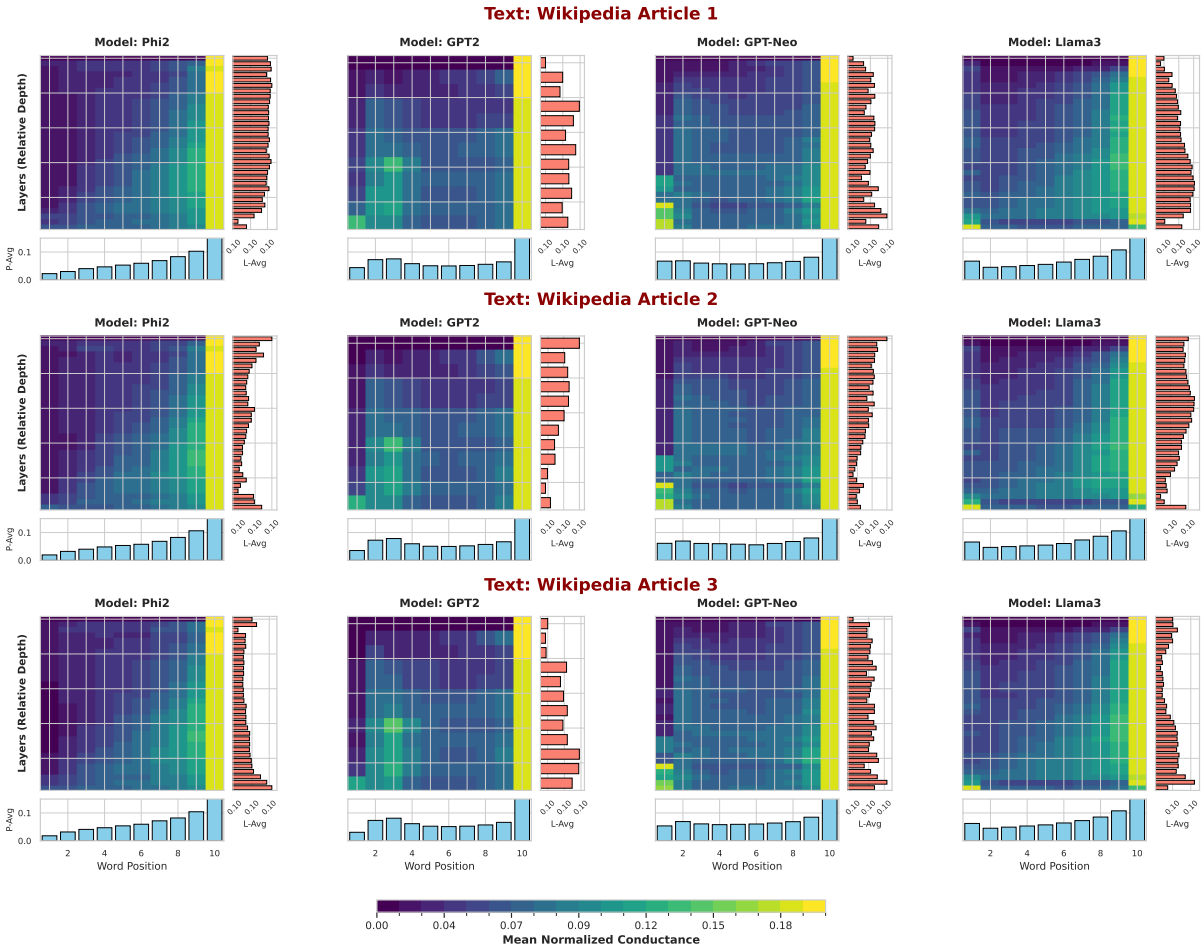

Figure A10: Layer-position heatmaps showing mean conductance. Mean conductance scores across layers (rows) and positions (columns) for each model and story.

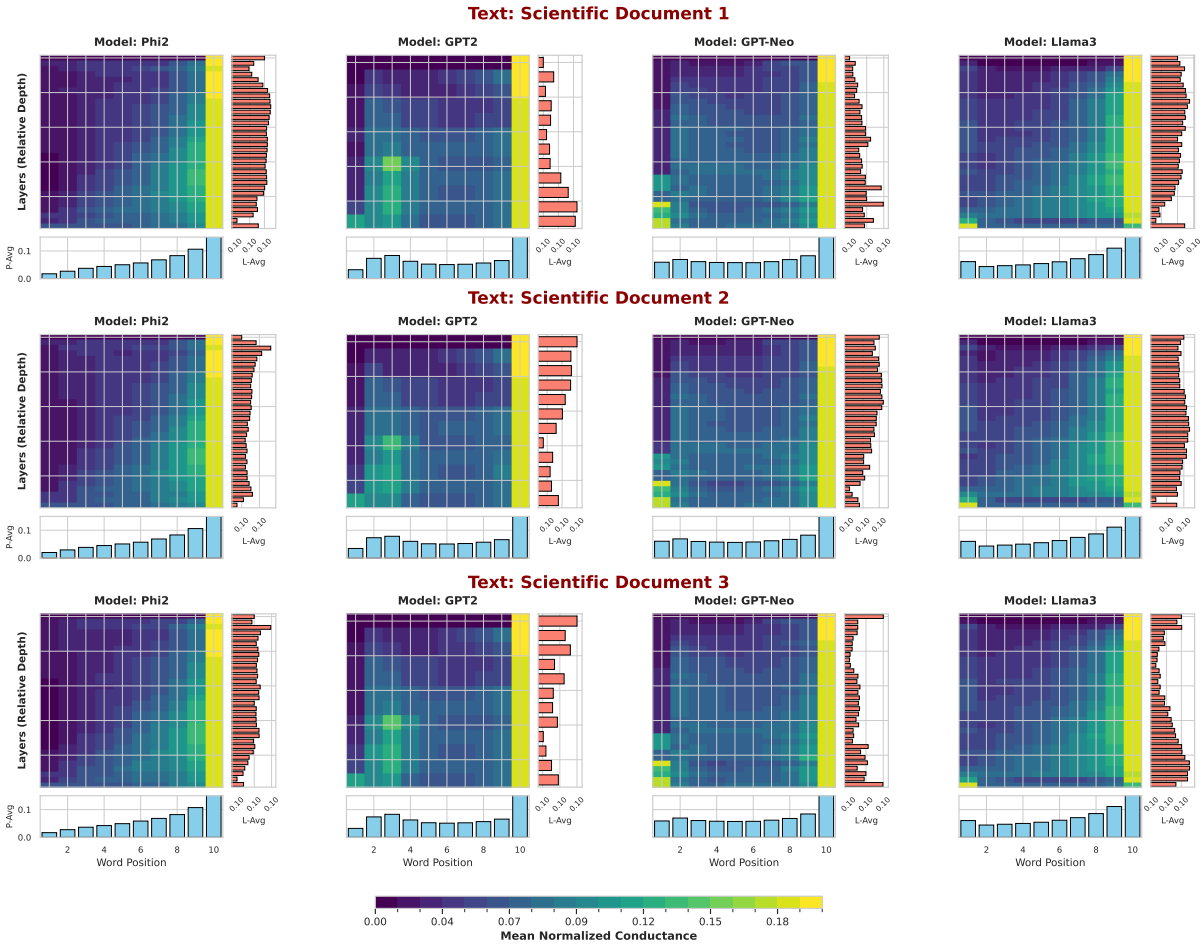

Figure A11: Layer-position heatmaps showing mean conductance. Mean conductance scores across layers (rows) and positions (columns) for each model and story.

C.2 Variance Heatmaps

Heatmaps visualize conductance across layers (rows) and positions (columns), providing an alternative view of positional importance patterns.

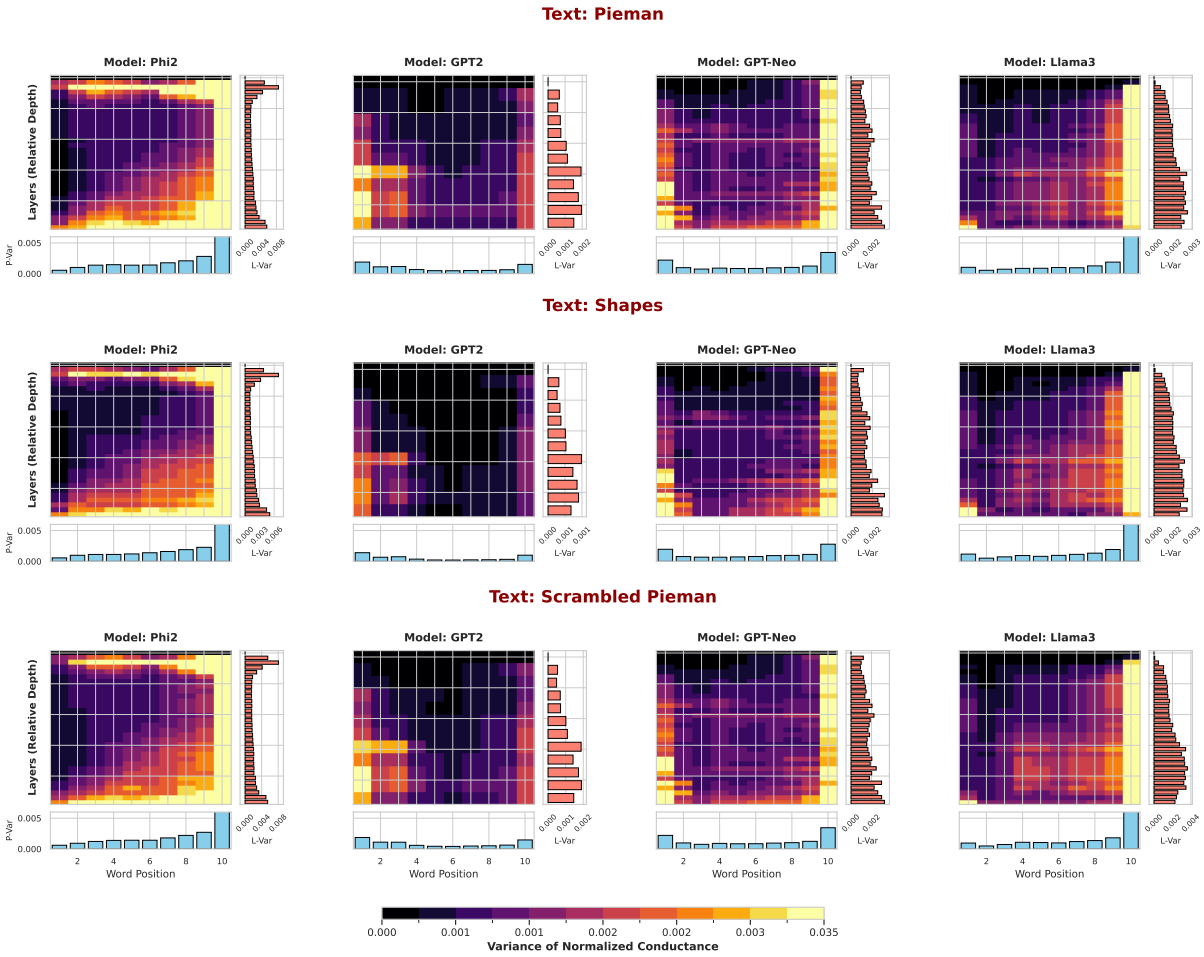

Figure A12: Layer-position heatmaps showing variance in conductance. Variance in conductance scores across layers (rows) and positions (columns) for each model and story. Shows positional stability of layer contributions.

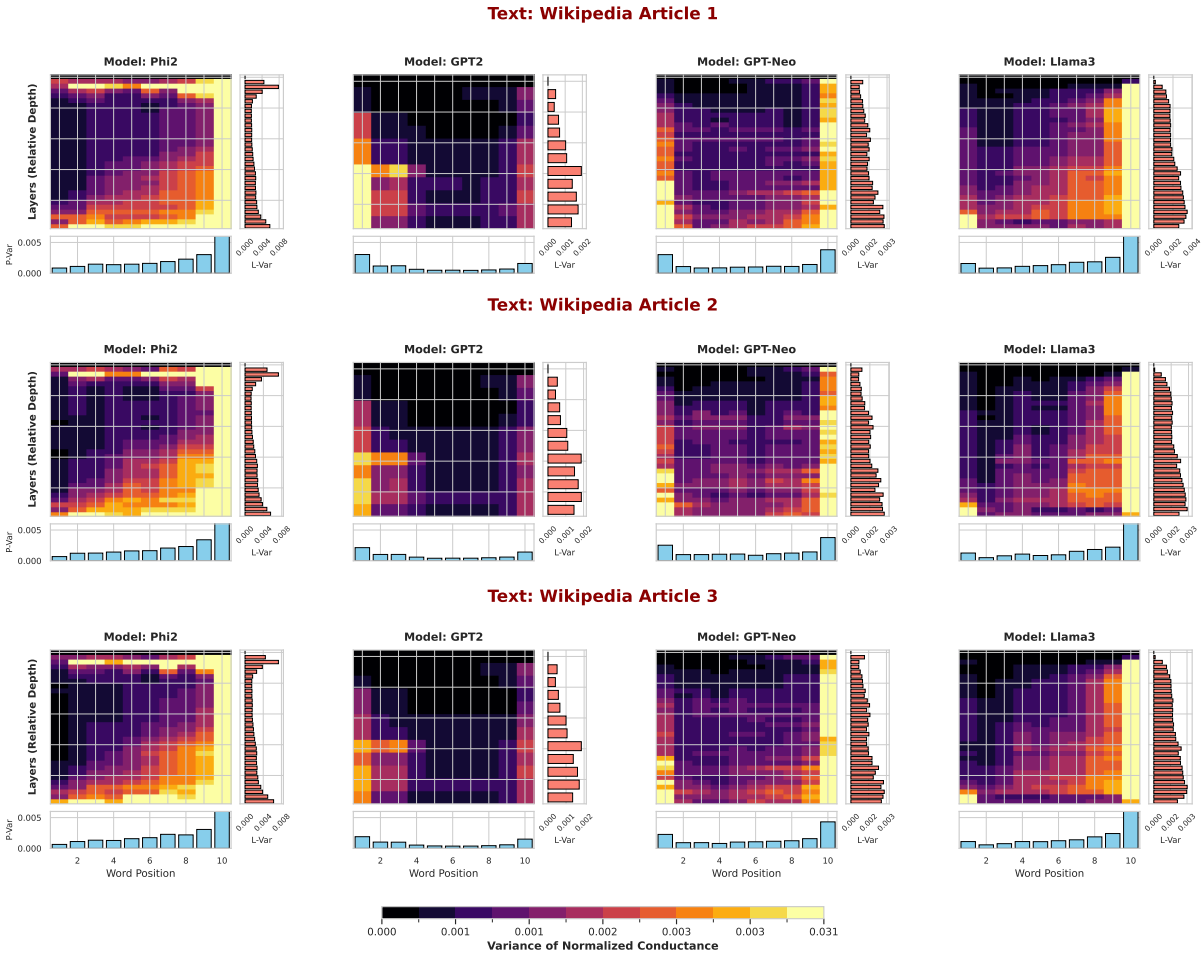

Figure A13: Layer-position heatmaps showing variance in conductance. Variance in conductance scores across layers (rows) and positions (columns) for each model and article.

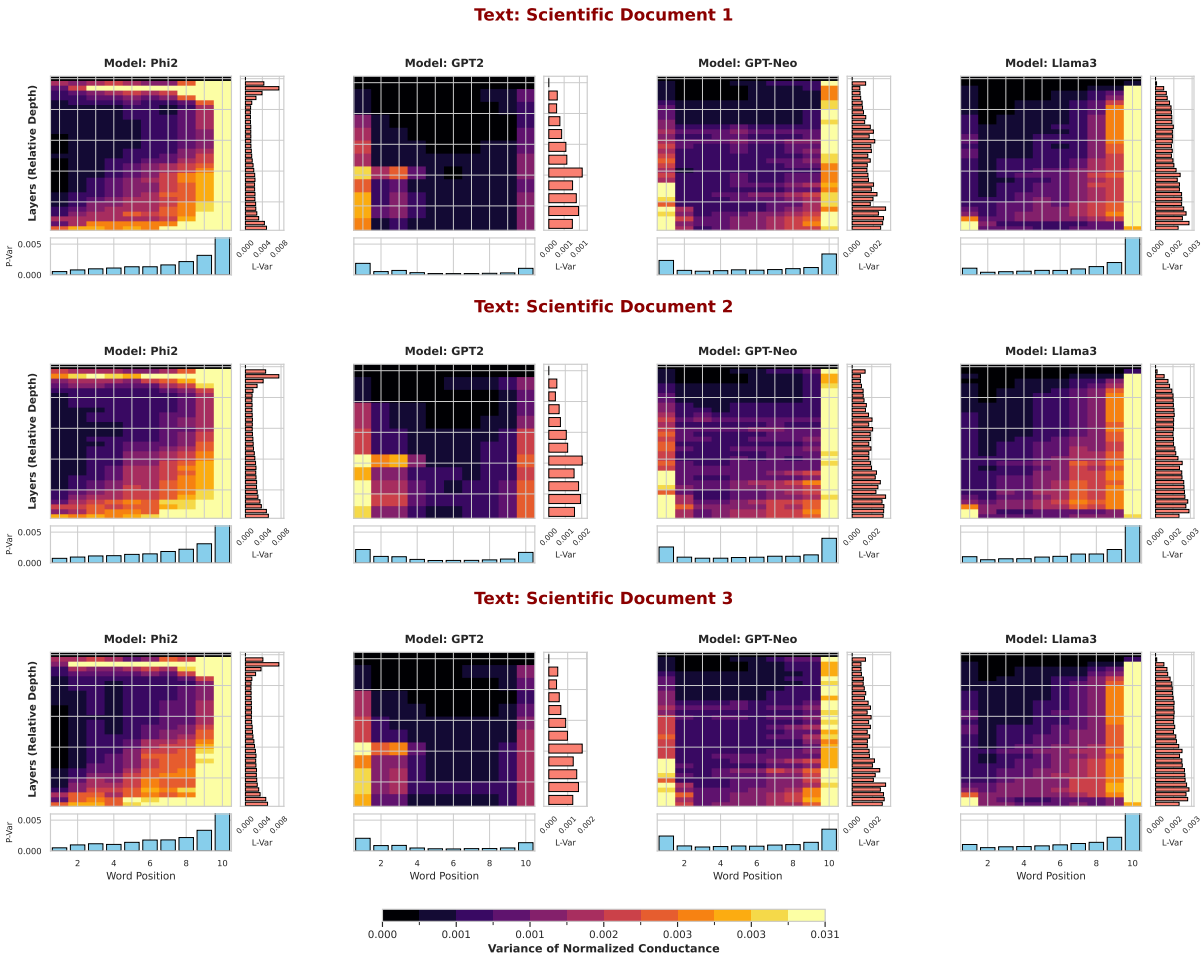

Figure A14: Layer-position heatmaps showing variance in conductance. Variance in conductance scores across layers (rows) and positions (columns) for each model and document.

D Layer Dominance Analysis by Text

D.1 Layer Dominance by Position

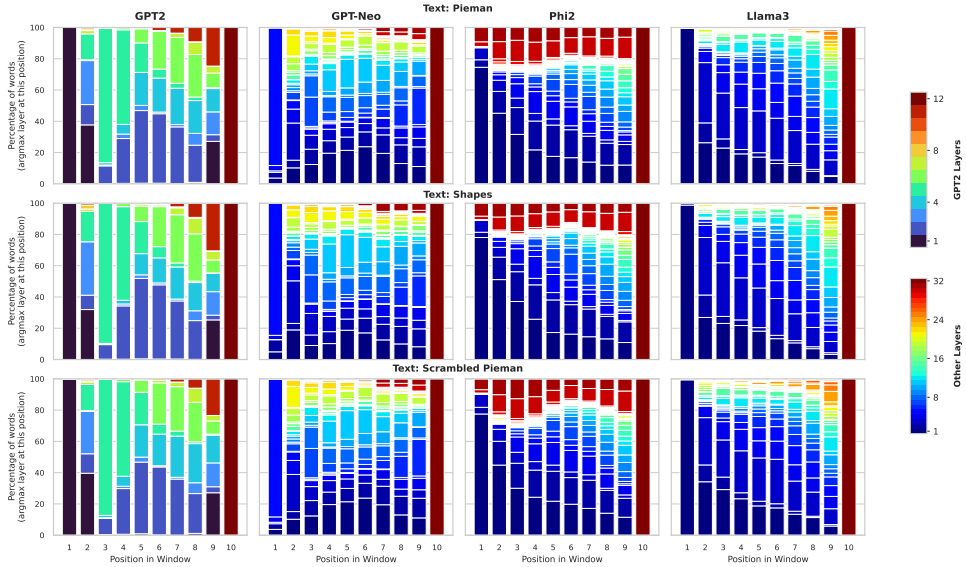

Figure A15: **Distribution of dominant layers across input positions for individual inputs.** Each panel shows the percentage of words dominated by each layer at each position. Rows show Pie Man (top), Shapes (middle), and Scrambled Pie Man (bottom). Columns show GPT-2, GPT-Neo, Phi-2, and Llama-3.

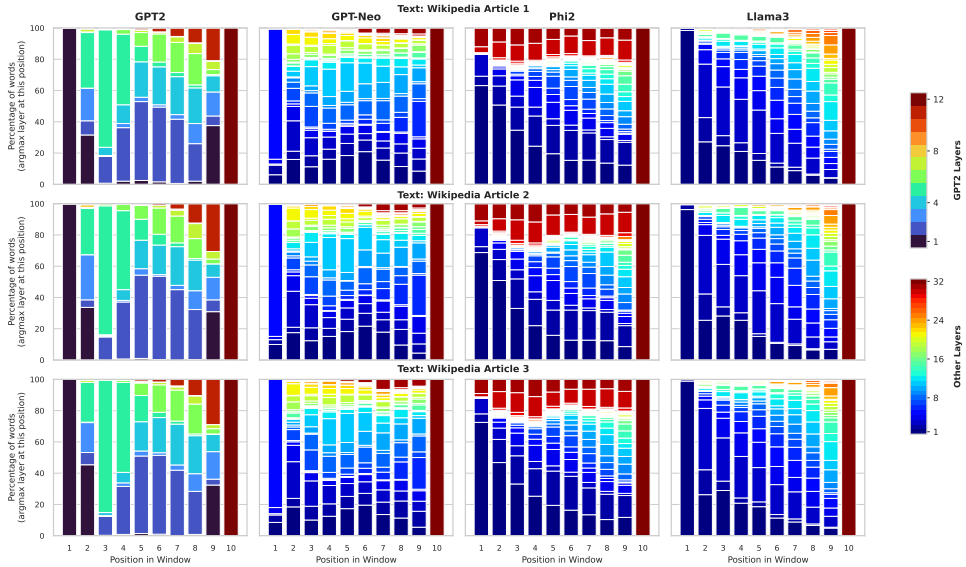

Figure A16: **Distribution of dominant layers across input positions for individual inputs.** Each panel shows the percentage of words dominated by each layer at each position. Rows show Wikipedia Articles. Columns show GPT-2, GPT-Neo, Phi-2, and Llama-3.

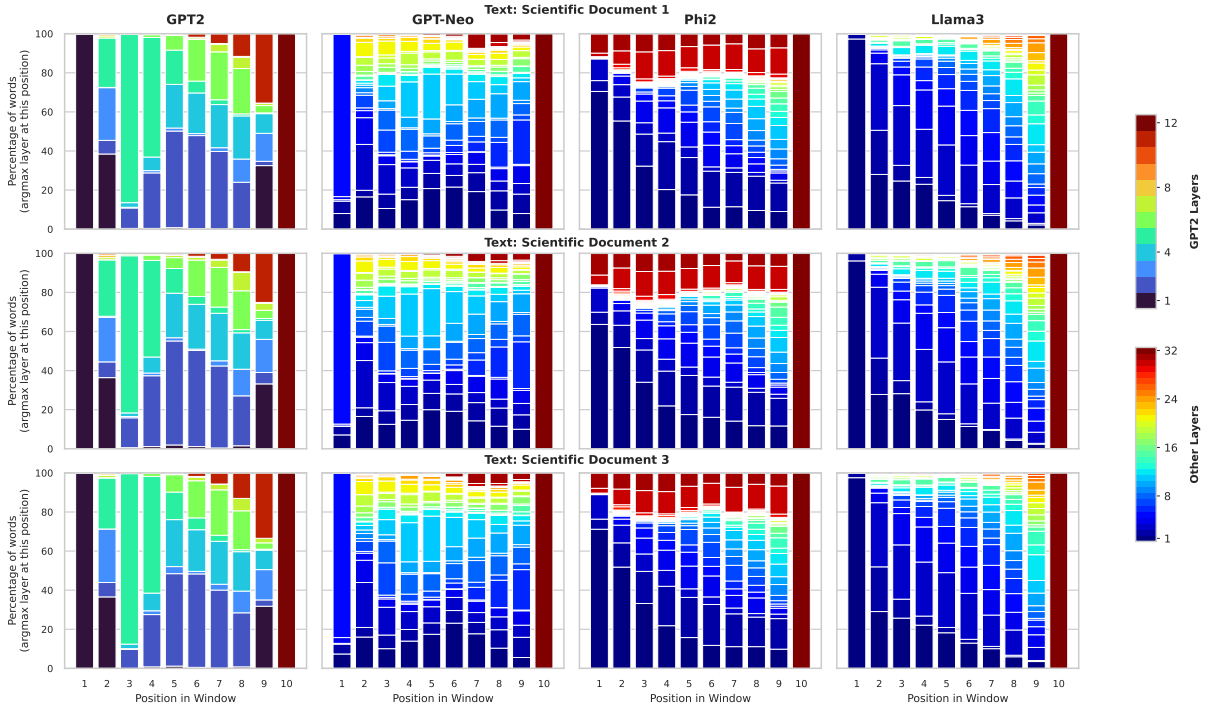

Figure A17: **Distribution of dominant layers across input positions for individual inputs.** Each panel shows the percentage of words dominated by each layer at each position. Rows show Scientific Documents. Columns show GPT-2, GPT-Neo, Phi-2, and Llama-3.

D.2 Cross-Text Consistency

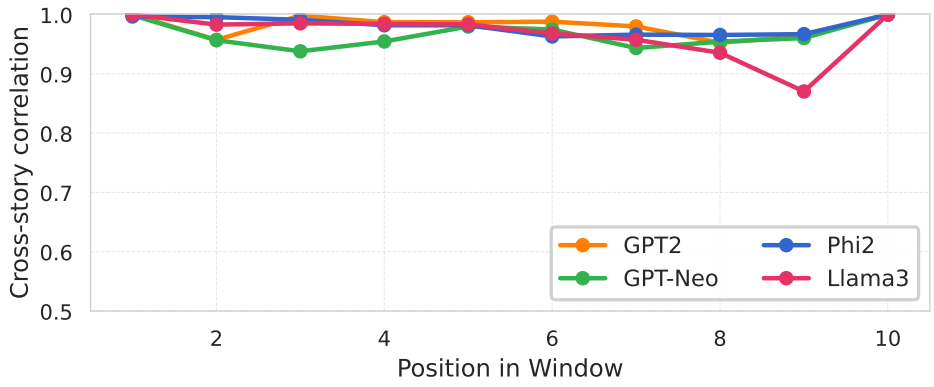

Figure A18: **Cross-story consistency of layer dominance patterns.** Mean pairwise Pearson correlation of layer dominance distributions. Correlations exceed $r > 0.8$, confirming architectural determination.

E Robustness to Window Length: Analysis with P=50

To verify that our findings are robust to window length, we repeated all analyses with $P = 50$ on two representative texts: Pie Man and Shapes. All key patterns replicate with the longer window, demonstrating that positional bias reflects intrinsic architectural properties independent of window length.

E.1 Positional Importance Profiles (P=50)

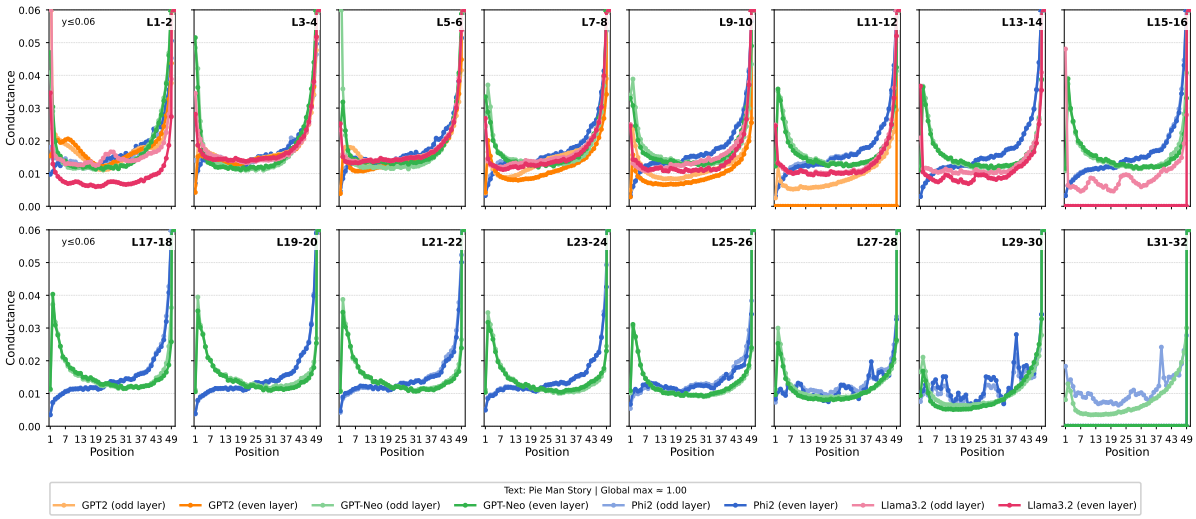

Figure A19: **Pie Man (P=50):** Layer-wise positional importance profiles. Patterns replicate those observed with $P = 10$ (main text Figure 2).

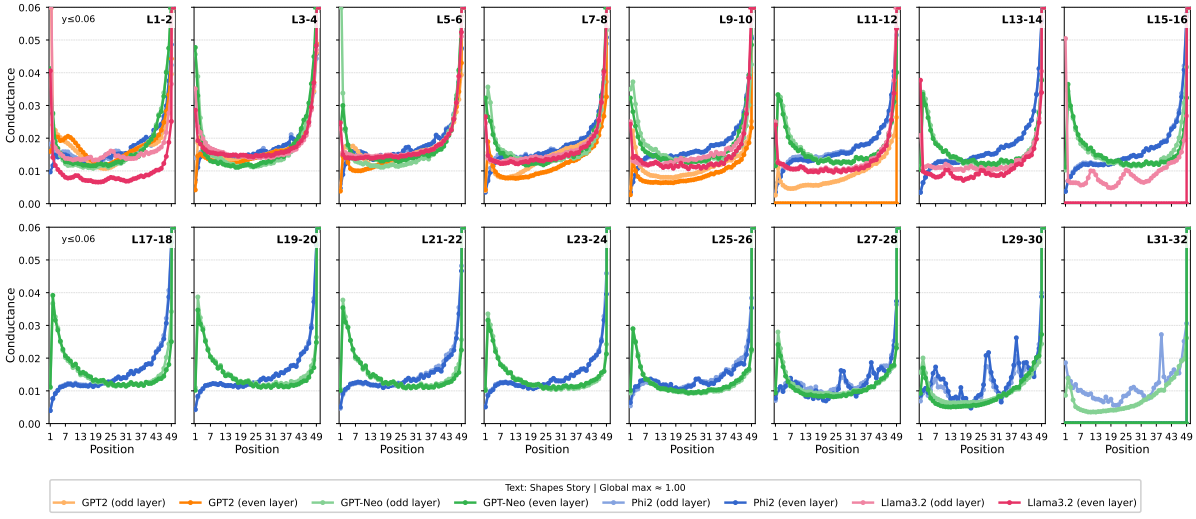

Figure A20: **Shapes (P=50):** Layer-wise positional importance profiles. Patterns consistent with $P = 10$ analysis.

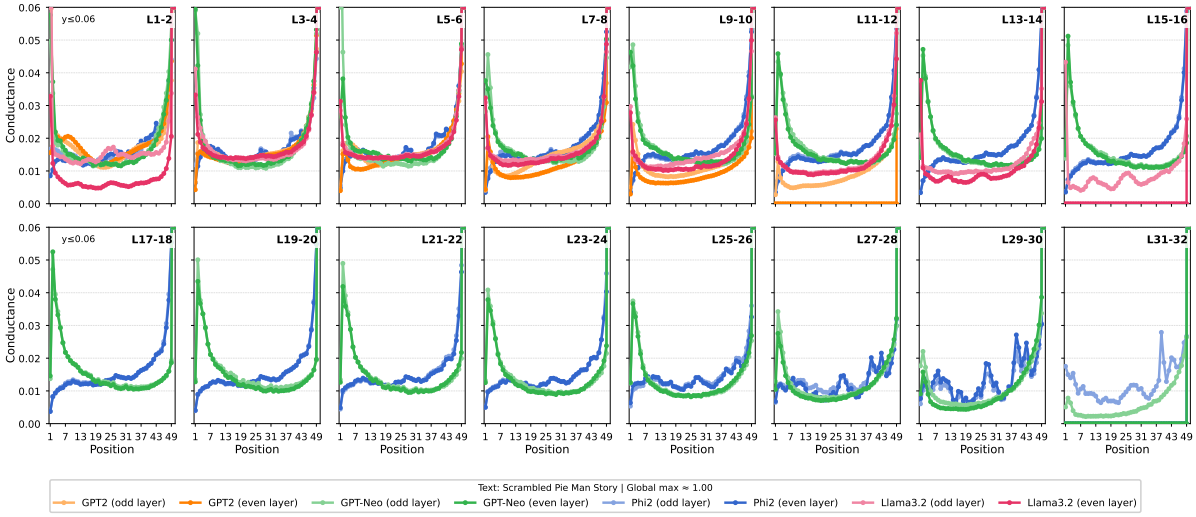

Figure A21: **Scrambled Pie Man (P=50):** Layer-wise positional importance profiles. Despite scrambling, positional patterns persist as with $P = 10$.

E.2 Cross-Text Consistency (P=50)

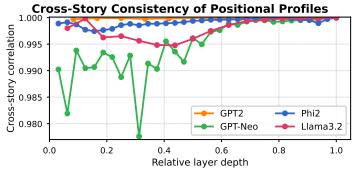

Figure A22: **Cross-text consistency (P=50).** Mean pairwise Pearson correlation between positional profiles across texts for each layer. Correlations exceed $r > 0.99$ across all models, replicating text-invariance with longer windows.

E.3 Primacy and Recency Evolution (P=50)

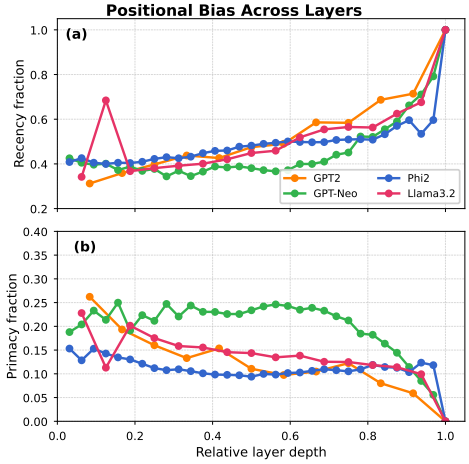

Figure A23: **Primacy and recency evolution (P=50).** (a) Recency fraction. (b) Primacy fraction. Patterns averaged across texts. Recency increases monotonically with depth; primacy remains weak, replicating $P = 10$ findings.

E.4 Word-Level Layer Importance (P=50)

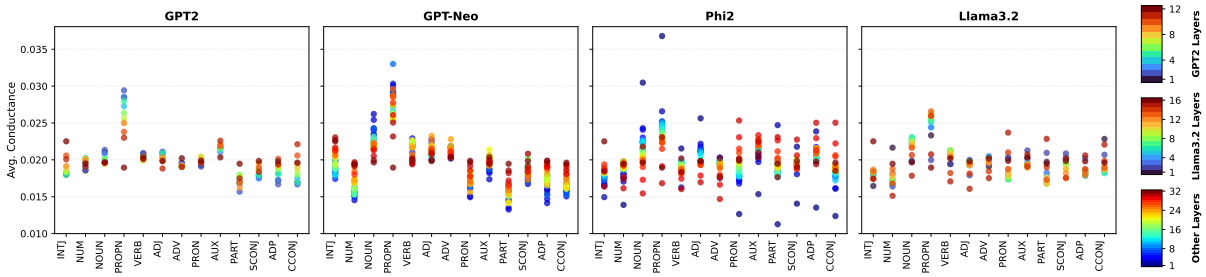

Figure A24: **Position-averaged importance by POS (P=50).** Layer conductance averaged over positions, aggregated by POS category. Content words receive higher importance in early layers, consistent with $P = 10$ analysis.

E.5 Layer Dominance (P=50)

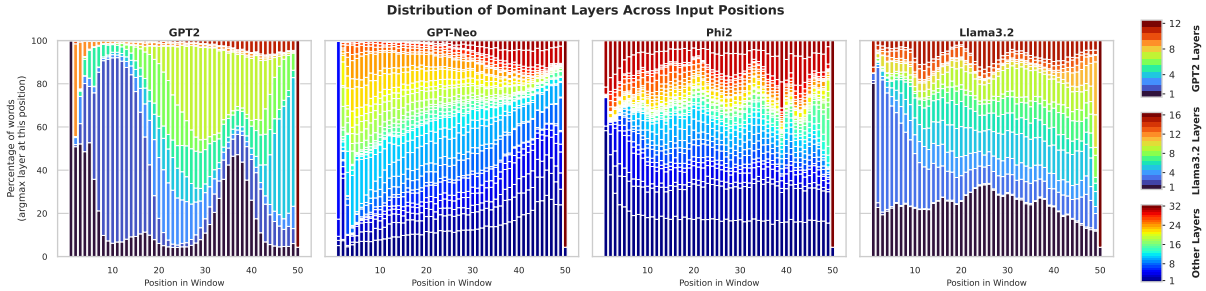

Figure A25: **Layer dominance by position (P=50).** Percentage of words at each position dominated by each layer. Early layers dominate most words at all positions, replicating $P = 10$ patterns.

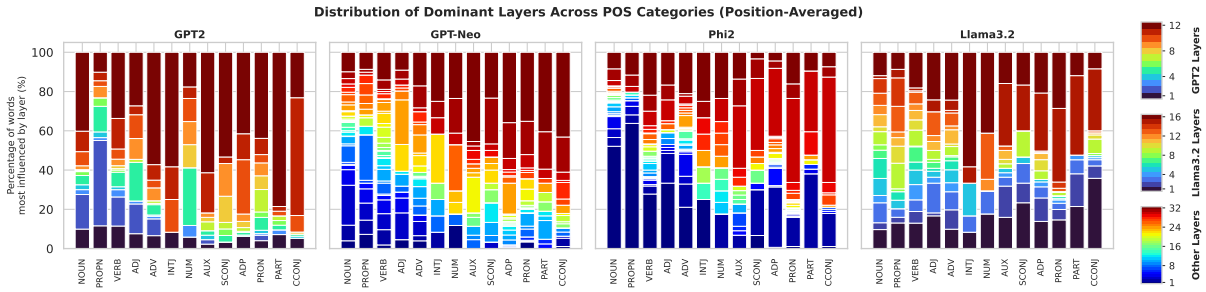

Figure A26: **Layer dominance by POS (P=50).** Percentage of words in each POS category dominated by each layer (position-averaged). Content words dominated by early layers; function words show more distributed dominance, consistent with $P = 10$.

E.6 Summary

All key findings from $P = 10$ analysis replicate with $P = 50$: text-invariant profiles ($r > 0.99$), monotonic recency increase with depth, weak and diminishing primacy, early layer dominance across positions, and word-type sensitivity in early layers.