The Remarkable Robustness of LLMs: Stages of Inference?

Vedang Lad*
MIT
Stanford University
vedanglad@stanford.edu

Jin Hwa Lee
University College London
jin.lee.22@ucl.ac.uk

Wes Gurnee
MIT
wesg@mit.edu

Max Tegmark
MIT
tegmark@mit.edu

Abstract

We investigate the robustness of Large Language Models (LLMs) to structural interventions by deleting and swapping adjacent layers during inference. Surprisingly, models retain 72–95% of their original top-1 prediction accuracy without any fine-tuning. We find that performance degradation is not uniform across layers: interventions to the early and final layers cause the most degradation, while the model is remarkably robust to dropping middle layers. This pattern of localized sensitivity motivates our hypothesis of four stages of inference, observed across diverse model families and sizes: (1) detokenization, where local context is integrated to lift raw token embeddings into higher-level representations; (2) feature engineering, where task- and entity-specific features are iteratively refined; (3) prediction ensembling, where hidden states are aggregated into plausible next-token predictions; and (4) residual calibration, where irrelevant features are suppressed to finalize the top-1 output distribution. Synthesizing behavioral and mechanistic evidence, we provide a hypothesis for interpreting depth-dependent computations in LLMs.

1 Introduction

Recent advancements in Large Language Models (LLMs) have exhibited remarkable reasoning capabilities, often attributed to increased scale [1]. Understanding these capabilities and mitigating associated risks [2–4] have motivated extensive research into their underlying mechanisms.

A *bottom-up* approach to interpretability, known as mechanistic interpretability, has explored the iterative inference hypothesis [5, 6], which posits that each transformer layer incrementally updates a token's hidden state toward minimizing loss by progressively shaping the next-token distribution [7]. This is supported by self-repair [6], where later layers correct or mitigate errors introduced by earlier layers, and redundancy [8, 9], where multiple layers perform similar or overlapping computations to refine predictions.

It remains unclear how this iterative view of inference fits with the "circuit" hypothesis, which argues for clearly delineated, specialized roles for certain model components. This is supported by induction heads [10], successor heads [11], copy suppression mechanisms [12], and knowledge neurons [13], among other "universal" neurons [14, 15]. Whereas iterative inference suggests gradual refinement through overlapping computations, the strong circuit hypothesis implies distinct, modular computational pathways.

*Corresponding author. Work started at MIT, completed at Stanford University

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

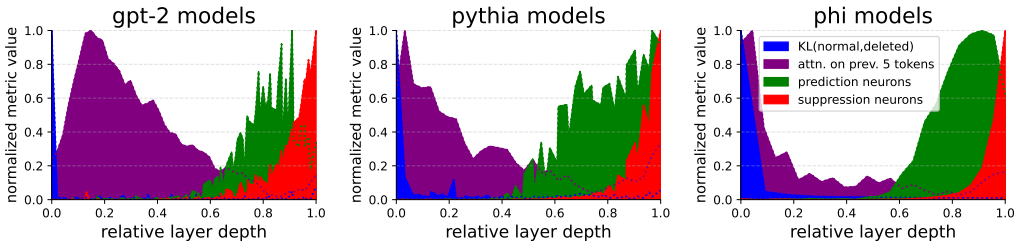


Figure 1: Statistical signatures of stages of inference averaged across three model families. (Blue) KL between the normal model and layer ℓ zero-ablated. (Purple) Total attention paid to the previous five tokens in a sequence. (Green) The number of "prediction" neurons (Red) The number of "suppression" neurons [20, 15, 14].

Table 1: Our Hypothesis: Stages of Inference

Stage	Name	Function	Observable signatures
1	Detokenization	Integrate local context to transform raw token representations into coherent entities	Catastrophic sensitivity to deletion and swapping and attention-heavy computation.
2	Feature Engineering	Iteratively build feature representation depending on token context	Little progress made towards next token prediction, but significant increase in probing accuracy and patching importance.
3	Prediction Ensembling	Convert previously constructed semantic features into plausible next token predictions using an iterative ensemble of model components	Prediction neurons appear and output distribution begins to narrow.
4	Residual Calibration	Eliminate obsolete features and form the next token distribution from internal representation	Suppression neurons appear and output distribution shapes for top-1 prediction with a growing MLP-output norm

Naturally, layer-wise phenomena in LLMs are also documented outside formal interpretability research and provide more evidence to existing interpretability findings. For example, while knowledge storage within mid-layer MLP neurons has been demonstrated [16], other non-interpretability work has found that fine-tuning predominantly affected the weights in the middle layers [17]. Quantization studies identified improved benchmark performance by retaining only low-rank MLP components from the middle to later layers [18]. Other works have noted a transition in activation sparsity from sparse to dense around mid-model depth [19, 15]. These behavioral findings, when integrated with mechanistic insights, suggest a layered computation structure not yet fully characterized.

To explore this structure, we perform layer-wise interventions—deleting individual layers or swapping adjacent ones (Figure 13)—to characterize their localized effects. Building on these insights, we analyze depth-wise roles and synthesize our findings with prior interpretability work to propose a four-phase hypothesis that attempts to bridge the top-down and bottom-up views of computation in decoder-only LLMs.

Concretely, we hypothesize four depth-dependent stages: **(1) detokenization**, **(2) feature engineering**, **(3) prediction ensembling**, and **(4) residual calibration**. In brief, early layers integrate local context to form coherent entities; middle layers iteratively construct features; later layers convert these features into next-token predictions via an ensemble of neurons. Figure 1 and Table 1 summarize these stages and their associated empirical signatures. We synthesize these findings with prior interpretability work [21] to suggest a depth-aligned computational structure in LLMs.

2 Related Work

Mechanistic Interpretability Mechanistic interpretability often employs circuit analysis to uncover model components relevant to specific computations. In computer vision, universal mechanisms such as frequency detectors and curve-circuits have been identified [22–24], with features become progressively more complex through the layers of CNNs. These principles were later extended to modern transformers [25, 26], where similar circuit-based analyses revealed phenomena such as circuit reuse [27], variable-finding mechanisms [28], self-repair [6, 29], function vectors [30, 31], and long-context retrieval [32].

Iterative Inference and Depth-Dependent Computations The iterative inference hypothesis, first explored in ResNets [33, 34], posits that each layer incrementally updates token representations. This idea has gained traction in transformers, particularly through logit lens analyses [35, 5], which visualize the model's evolving prediction distributions layer by layer. Some studies further suggest discrete inference phases [36], with certain computations localized to specific depths—such as truth-processing [37] or multilingual translation [38]. These findings are complemented by layer permutation studies showing that performance improves when self-attention layers precede feedforward layers [39].

BERTology Prior work on ablations and layer-wise analysis has primarily focused on BERT [40]. These studies reveal substantial redundancy: even with aggressive neuron and layer pruning, models retain most of their performance [41–45]. More recent investigations corroborate this, showing that a significant portion of attention heads and feedforward components can be removed with minimal accuracy loss [9, 8].

3 Experimental Protocol

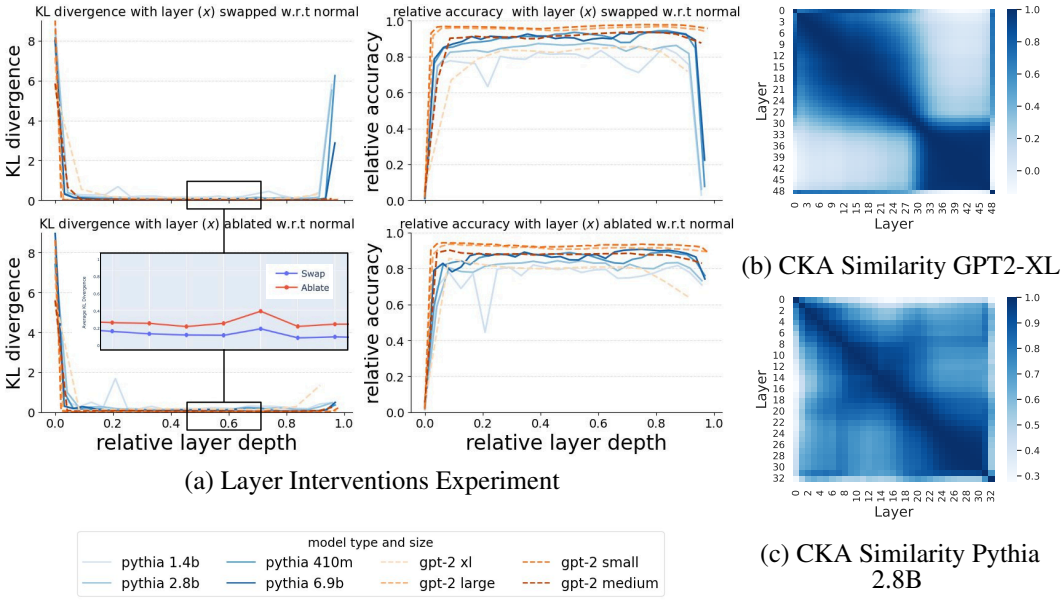
Table 2: Comparison of Language Model Architectures

Model Series	Size	Layers
Pythia	410M	24
	1.4B	24
	2.8B	32
	6.9B	32
	12B	36
GPT-2	Small (124M)	12
	Medium (355M)	24
	Large (774M)	36
	XL (1.5B)	48
Microsoft Phi	Phi-1 (1.3B)	24
	Phi-1.5 (1.3B)	24
	Phi-2 (2.7B)	32
Llama 3.2	1B	16
	3B	28
Qwen 2.5	0.5B	24
	1.5B	28
	3B	36
	14B	48

Models To investigate the stages of inference in language models, we examine the Pythia [46], GPT-2 [47], Qwen 2.5 [48], LLaMA 3.2 [49], and Microsoft Phi [50, 51] model families, which range from 124M to 14B parameters (see Table 2). All families use decoder-only transformers but differ in their execution of attention and MLP components. Specifically, Pythia models execute attention and MLP layers in parallel. In contrast, GPT-2, Phi, and Llama models apply attention followed by an MLP sequentially. We preprocess weights identically across all models, folding in the layer norm, centering the unembedding weights, and centering the writing weights as described in Appendix B. Despite these architectural differences, most phenomena remain consistent across models, though we discuss drawbacks in Limitations 6.

Data Besides data agnostic experiments, we evaluate all five model families on a corpus of one million tokens from random sequences of the Pile dataset [52], unless otherwise noted in the experiment.

Layer Swap Data Collection To study the robustness and role of different model components at different depths, we employ a swapping intervention where we switch the execution order of a



(a) Layer Interventions Experiment

(b) CKA Similarity GPT2-XL

(c) CKA Similarity Pythia 2.8B

Figure 5: (a) Effect of layer swap (top) and layer drop (bottom) interventions on model behavior. (left) KL divergence between the intervened and original models. (right) Consistency of top-1 predictions. (b)(c) Representational similarity across layers measured using CKA, showing block-like structure in GPT-2 XL (b) and Pythia 2.8B (c). Similar trends are observed across other model families and sizes (see Appendix C).

pair of adjacent layers in the model. Specifically, for a swap intervention at layer ℓ, we execute the transformer block (including the attention layer, MLP, and normalization) $\ell + 1$ before executing block ℓ. We record the Kullback-Leibler (KL) divergence between the intervened and original models output distribution, along with the loss, top-1 prediction accuracy, prediction entropy, and benchmark task performance. This intervention allows us to examine how the order of computation affects the model's behavior and performance at different depths.

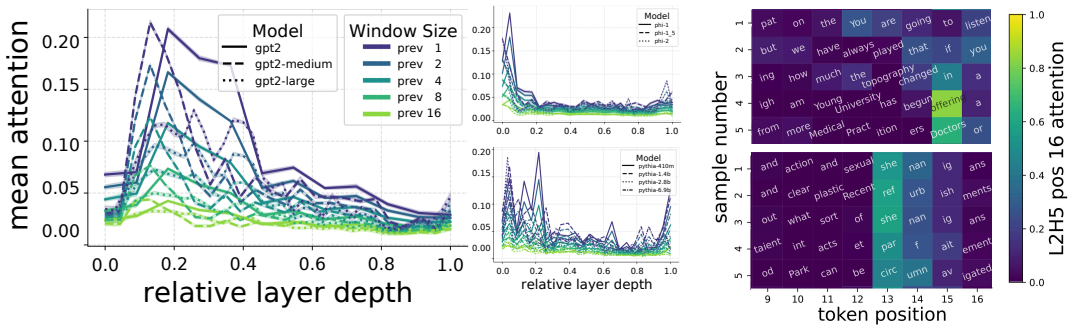
Ablation Data Collection To generate baselines for each layer swap experiment, we perform zero ablations on the corresponding layer while collecting the same metrics. The ablation preserves the swap ordering: for a swap ordering of **1-2-4**-3-5, the ablation maintains **1-2-4**-5. We opt for zero ablation as opposed to mean ablation, as proposed by [5], to maintain consistency with the swap order.

4 Robustness

4.1 Intervention Results

We apply our aforementioned drop and swap interventions to every layer of four GPT-2 models [53] and four Pythia models [46]. In Figure 5, we report (1) the KL divergence between the prediction of the intervened model and the nominal model, (2) the fraction of predictions that are the same between the intervened model and the baseline model (denoted as relative accuracy). We also report the performance on common benchmark tasks (HellaSwag[54], ARC-Easy[55] and LAMBADA[56]) for all models in Figure 15-16, which show a similar trend.

In contrast to the first and last layers' interventions, the middle layers are remarkably robust to both deletion and minor order changes. When zooming in on the differences between the effect of swaps and drops for intermediate layers, we find that swapping adjacent layers is less harmful than ablating layers, matching a result in vision transformers [26]. We take this as an indication that certain operations within the forward pass are commutative, though further experimentation is required.



(a) (b)

Figure 6: (a) The average (across heads within a layer and query tokens) attention weight placed on the preceding 1, 2, 4, 8, 16 tokens for each layer. (b) Attention from the source token to the final token in various inputs. An identified sub-word merging attention head (bottom) found in the early layers of language models is responsible for attending to multi-token words (i.e, shenanigans, refurbishments, parfaitement, circumnavigate), compared to the baseline set of random non-multi-token words (top).

Intervening on the first layer is catastrophic for model performance for every model, regardless of size or model family. Specifically, dropping or swapping the first layer causes the model to have very high entropy predictions as opposed to causing a mode collapse on a constant token. In some models, swapping the last layer with the second-to-last layer also has a similar catastrophic high-entropy effect, while GPT-2 models largely preserve their predictions. This phenomenon motivates our study into the first few layers of the model, specifically the role paid by attention heads in these layers.

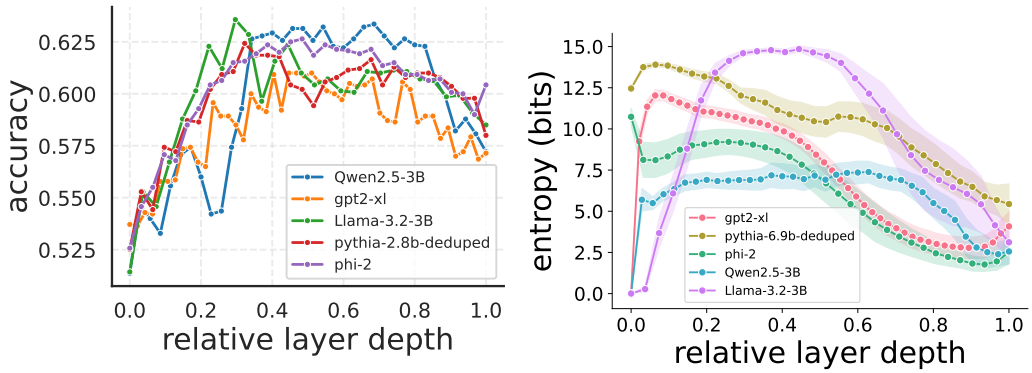
5 Stages of Inference Hypothesis

Motivated by the distinct phenomena at the first few and final few layers, we measured representational similarity across each layer output using Centered Kernel Analysis (CKA)[57–59]. This revealed a block-like structure across multiple models as shown in Figure 4. The existence of blocks reflects the robustness observed in the layer-wise intervention. Furthermore, the depth-dependent phase structure indicates that a shared computation motif across adjacent layers occurs in stages.

5.1 Stage 1: Detokenization

Given the extreme sensitivity of the model to first-layer ablations, we infer that the first layer is not a normal layer, but rather an extension of the embedding. Uniquely, the first layer is the layer that moves from the embedding basis to that of the transformer's residual stream. It is *only* a function of the current token. Consequently, by ablating the first layer, the rest of the network is blind to the immediate context and is thrown off distribution. Immediately after computing this extended embedding, evidence from the literature suggests that the model concatenates nearby tokens that are part of the same underlying word [60, 61] or entity [62] (e.g., a first and last name). This operation integrates local context to transform raw token representations into coherent entities. In this way, the input is "detokenized" [36, 63]. Previous work has shown the existence of neurons that activate for specific n-grams [63, 15]. Of course, to accomplish this, there must be attention heads that copy nearby previous tokens into the current token's residual stream.

Sub-word Merging Heads To further examine this detokenization mechanism, we investigated attention heads responsible for constructing multi-token words, known as *sub-word merging* heads [61]. These heads help capture the context of a token for appropriate prediction, thus contributing to the detokenization process. We constructed a dataset with two classes: each consisting of 16 tokens, where in one class, the final 4 tokens form a word. Our analysis identified specific heads in the early layers of models that contribute solely to the construction of these multi-token words. As illustrated in Figure 6b, layer 2 head 5 of Pythia 2.8B moves information from earlier tokens to the final token of the word. The attention heads exhibit a consistent pattern, where attention decreases as tokens approach the final word. Specifically, the final token of the word attends most strongly to the first



(a) (b)

Figure 7: (a) Layer-wise probe accuracy on contextual lexical meaning (WiC task), peaking in intermediate layers is suggestive of where semantic features are linearly encoded. (b) Using the logit lens technique [35], we calculate the probability distribution of the next token at the end of every layer, and then take its entropy. This provides a measure of the model's confidence in the next prediction. Despite high probe accuracy, the residual, but high entropy residual stream suggests that semantic features exist mid-model but are not yet used for prediction. For all models see Appendix 18 and 19.

token, a feature absent in the baseline. This suggests at least one of many mechanisms by which models integrate local context, occurring at higher density in the first half of the models.

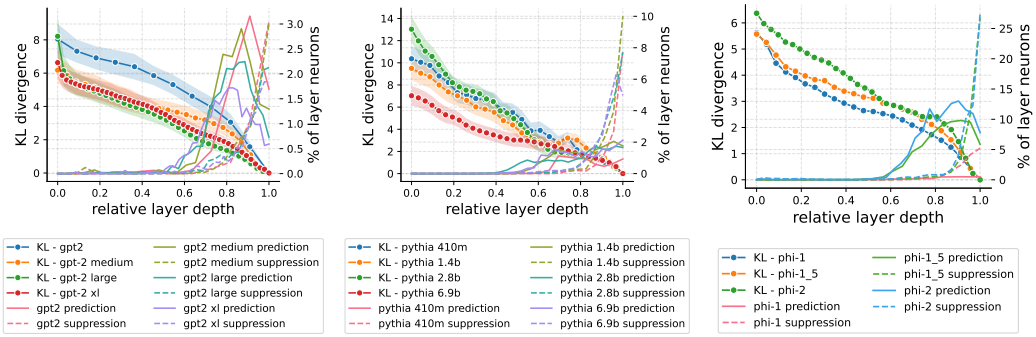
Local Attention If early layers indeed specialize in integrating local context, then we would expect attention heads in these layers to disproportionately focus on tokens close to the current position. To investigate this hypothesis, we measure the fraction of attention that each token directs toward preceding tokens at varying distances. As shown in Figure 6, attention heads in early layers are strongly biased towards nearby tokens, with attention becoming progressively less localized in deeper layers.

5.2 Stage 2: Feature Engineering

After integrating local context in the early layers—e.g., stitching together sub-word tokens and forming short-range dependencies—the model must begin converting those localized representations into more semantically meaningful features. We hypothesize that this marks the beginning of a feature engineering stage, in which the model constructs intermediate features that encode abstract properties useful for downstream prediction.

Prior work provides indirect support for this idea. Model editing studies suggest that factual information is stored in mid-layer MLPs [16, 64, 62], while probing experiments have found that intermediate layers encode features related to sentiment [65], truth [37], and temporal structure [66]. These studies typically show that probing accuracy rises through the early layers, peaks near the midpoint, and then declines, suggesting that features are constructed and later transformed or compressed. Related work also observes a shift from syntactic to semantic representations with depth [36, 38].

WiC Probing To illustrate this pattern, we train linear probes to detect context-dependent word meaning using the WiC (Word-in-Context) task [67, 68]. For instance, given two sentences containing the word bank, the task is to classify whether it is used with the same meaning. Examples include distinguishing "the river *bank*" from "the robbed *bank*," where the same word has different meanings depending on the context. We apply this probe at each layer of the model, using the hidden state of the target word in context. As shown in Figure 7 (left), the accuracy of the probe increases through the early layers, peaks in the middle of the model, and then decreases, supporting the hypothesis that semantic features are most *linearly* accessible in the intermediate layers. We extend the observation across model families and sizes in Figure 18.



(a) GPT models (b) Pythia models (c) Phi models

Figure 8: We measure KL divergence between intermediate and final predictions using the logit lens method [35]. On the second axis, we use an automated procedure for classifying neuron types detailed in [14], into prediction neurons and suppression neurons. These are universal neurons in all models known to increase the probabilities of tokens and decrease the probabilities of others. We hypothesize this inverse relationship as evidence for ensembling in networks[15].

Logit Lens While these results suggest that intermediate representations encode semantic information, it remains unclear whether such features contribute to prediction at this stage. To investigate this, we apply the logit lens [35, 69], which projects the residual stream at each layer into the output vocabulary space using the model's unembedding matrix. This provides a layer-wise estimate of the model's next-token distribution.

We compute both the entropy of the intermediate predictions and their KL divergence from the model output. As shown in Figure 7 (right), entropy remains high and KL divergence low throughout the early and middle layers. In other words, while meaningful features appear to be present in the residual stream at this stage, the model's output distribution remains high in entropy, indicating that these features have not yet been consolidated into confident next-token predictions. Bridging this gap requires a mechanism that selectively retains information from relevant features while filtering out irrelevant ones, thereby reducing uncertainty in the output distribution.

5.3 Stage 3: Prediction Ensembling

Around the midpoint of the model, we observe a qualitative shift in behavior. Having constructed semantic features in the earlier layers, the model must begin converting these into specific next-token predictions. Evidence for this transition comes from the logit lens, where we observe a steady decline in entropy (Figure 7 right) and KL divergence (Figure 8) between intermediate and final predictions beginning around the middle layers. This suggests that the model is gradually committing to a particular output, aggregating semantic features into a more concrete distribution over tokens.

This region of the model also displays high robustness to layer interventions (Figure 5), suggesting redundancy or capacity for self-repair. One possible cause of this resilience is the presence of overlapping computational pathways [6, 70]. Rather than relying on a single deterministic path, the model seems to combine multiple signals—both across and within layers—to form its prediction. We explore this mechanism by identifying the neurons that contribute to prediction, testing their collective behavior through a case study, and analyzing their distributional effects across depth.

Ensembling Within these overlapping pathways, we investigate specialized ensembles known as prediction neurons—units that systematically promote the likelihood of specific tokens [15, 7, 14]. These neurons work in tandem with suppression neurons (discussed in Section 5.4) to shape the model's output.

Prediction and Suppression neurons Following previous work[14], we identify these neurons by analyzing the MLP output weights $\mathbf{w}_{out}$ and their projection into vocabulary space via the unembedding matrix $\mathbf{W}_U$. Prediction neurons exhibit a logit effect distribution $\mathbf{W}_U \cdot \mathbf{w}_{out}$ with high kurtosis and positive skew; suppression neurons exhibit high kurtosis and negative skew. Across 18

models, prediction neurons begin to appear around the midpoint, increasing in density through the latter layers (Figure 8), before being overtaken by suppression neurons. For a detailed analysis of the detection and characterization of prediction, suppression, and other "universal" neurons, we refer readers to the original work [14].

Probing for the Suffix "-ing" We hypothesize that ensembles of prediction and suppression neurons collectively support next-token prediction. To test this, we construct a balanced classification task: given a 24-token context to a verb, does the final token end with or without "-*ing*"? We train linear probes on the activations of 32 high-variance prediction and suppression neurons, both individually and in groups. Neurons are selected using the criteria above, and examples from GPT-2 XL are shown in Figure 9. The full neuron list appears in Appendix 21.

We train two types of probes on the penultimate token's activations: 32 individual neuron probes and top-k ensemble probes ranked by individual accuracy (Figure 9). Suppression neurons yield the strongest individual probes, performing on par with the model's predictions (dotted red line). Ensemble probes trained on prediction neurons outperform both individual neurons and the model average, suggesting an important interplay between the two neuron types.

Density Effects The balance between prediction and suppression neurons appears to shape the model's output. To test this, we analyze how their density relates to the KL divergence between each layer's logit lens distribution and the final output. The sharpest decline in divergence corresponds closely with the rise in prediction neuron density, which peaks at roughly 85% of model depth.

Model comparisons further reinforce this pattern. Phi-1 has fewer prediction neurons and a shallower KL slope compared to later Phi models (Figure 8c). GPT and newer Phi models show steeper, smoother KL divergence drops than Pythia (Figures 8a, 8b). Notably, the most performant Phi models exhibit nearly 15% prediction and 25% suppression neurons per layer—5–8$\times$ the density in GPT-2 and 3–7$\times$ that of Pythia.

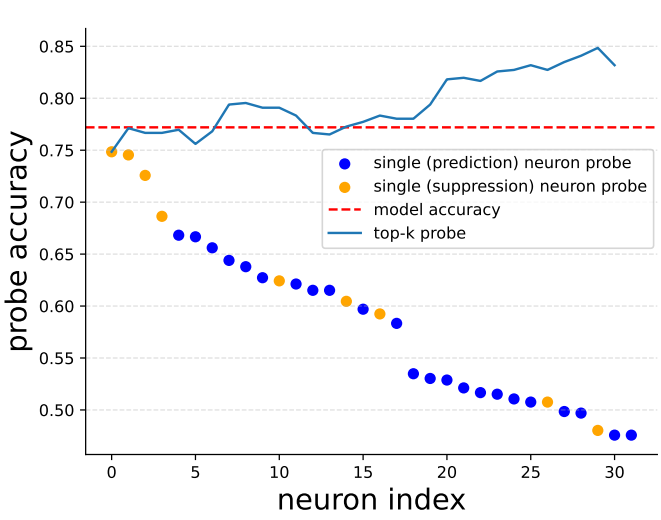
Interestingly, the density of prediction neurons decreases near the final 10% of layers, even as the model continues to converge on its output, sometimes accelerating(Figure 8b). This suggests the involvement of a distinct final-stage mechanism, which we delineate as a separate stage.

5.4 Stage 4: Residual Calibration

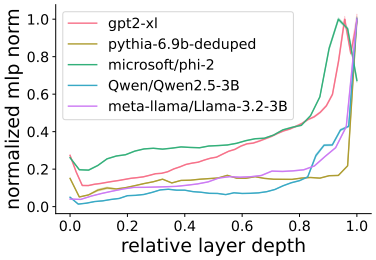
As prediction neuron density declines in the final layers, a different mechanism emerges. Across all models, we observe a sharp rise in suppression neurons near the end of the network. This transition from prediction to suppression neurons frequently coincides with an inflection point: entropy stops decreasing and begins increasing in the final layers (Figure 19b). Unlike prediction neurons, which promote likely tokens, suppression neurons refine the model's output by removing obsolete features and down-weighting improbable tokens. The resulting entropy increase in the final layers suggests that suppression neurons serve to calibrate the model's output toward the task it was trained for: producing a well-formed distribution over possible next tokens.

Layer Repeating Experiment To further explore this hypothesis, we design an experiment where we *repeat* certain layers of the model. Specifically, we duplicate blocks of layers within the model—for example, repeating layers 5 through 7 results in a sequence like (...4-5-5-6-6-7-7-8-9...). For this analysis, we fix the number of repeats to 1 and the block length to 5 (see additional results across model sizes and block length in Figure 25,23). In Figure 11, we observe that repeating blocks in the latter half of the model leads to a consistent decrease in entropy relative to the baseline (horizontal line). When evaluated on downstream benchmarks, the models with repeated layers at the last 80-90% of depth also exhibit improved performance on benchmarks, suggestive of residual calibration and the late-stage influence of prediction and suppression neurons. (Appendix 24).

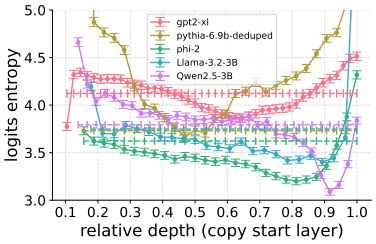
Final Layer The intensity of suppression neurons, as seen in Figure 8, is localized in the final few layers of the model, where the quantity of suppression neurons outstrips prediction neurons. To quantify the *intensity* of this change to the output distribution, we measure the norm of the MLP output, where a larger norm suggests a greater contribution to the residual (Figure 10). This also coincides with an increase in entropy (Figure 19b).



(a) Probing For Neuron Ensembles in GPT2-XL



(b) MLP Output Norms



(c) Layer Repeating Experiment

Figure 12: (a) Accuracy of linear probes trained to predict whether the final token ends in "-ing," using activations from individual prediction and suppression neurons (scatter points) and ensembles of neurons (blue line). Ensembles outperform individual probes and occasionally exceed the model's top-1 accuracy (red dotted line), consistent with the presence of "prediction ensembling." (b) Layer-wise MLP output norms across all 18 models show a rise toward the final layers, suggesting increasing residual contribution late in the model. (c) Repeating layers from the later half of a model reduces final-layer logit entropy more than repeating earlier layers or using the original model (dotted line), suggestive of residual calibration and the late-stage influence of prediction and suppression neurons.

6 Concluding Remarks

Why Are Language Models Robust to Layer-Wise Interventions? We hypothesize that the robustness of language models to layer deletion and swapping stems in part from the transformer's residual architecture. This interpretation aligns with our findings on prediction and suppression neurons: multiple computational pathways appear to contribute to the same output, allowing the network to tolerate disruption in any single path. The residual stream promotes this "ensembling", enabling gradient descent to construct shallow sub-networks that can operate in parallel. This architectural flexibility reduces the model's reliance on any specific layer, explaining its resilience to local interventions and supporting observed self-repair behavior and overlapping representations.

Limitations and Future Work While our four-stage hypothesis captures broad, depth-dependent patterns in LLMs, several caveats remain. Stage boundaries are approximate, and multiple stages may co-occur within a single layer. The findings reflect aggregate trends, whereas individual tokens may follow distinct processing paths. Additionally, we do not isolate the factors behind model-specific differences; e.g., GPT's greater robustness could arise from dropout, architectural variations, or depth. These limitations point directly to promising directions for future research. Future work should seek to sharpen these boundaries, link them to optimization dynamics, and test this hypothesis with a theoretical account to explain the empirical results.

Conclusion This work introduces a four-stage hypothesis for understanding inference in large language models, grounded in a diverse set of behavioral and mechanistic analyses. By examining how models respond to structural interventions—layer deletion and swapping—as well as probing attention patterns, neuron function, and residual stream dynamics, we identify a repeatable depth-wise structure to model computation. These stages—detokenization, feature engineering, prediction ensembling, and residual calibration—emerge across architectures and scales, suggesting that transformers perform inference not as a flat pipeline but as an ordered composition of specialized computational regimes. While not definitive, the strength and consistency of the empirical signatures presented here provide compelling evidence in support of the proposed hypothesis. Rather than aiming for an exhaustive

mechanistic dissection, we offer a unifying perspective that synthesizes and extends prior findings in and outside the interpretability literature.

More broadly, this layered view of inference has implications for how we interpret, audit, and intervene on language models. Understanding not just what a model computes, but when and where it computes it, may inform future approaches to alignment, compression, and modularity in model design. We hope this hypothesis serves as a foundation for a deeper investigation into the emerging capabilities of LLMs.

Contributions and Acknowledgments VL conceived and led the study, performed the analyses, and drafted the paper. JL conducted the sharpening experiment, WiC analysis, benchmarking, and CKA experiments. JL, WG and MT contributed to experimental design and analytical methodology, provided critical revisions, and WG and JL assisted with writing. A big thanks to Katrin Franke, Surya Ganguli, Sophia Sanborn, Eric Michaud, Josh Engels, Dowon Baek, Isaac Liao for helpful feedback.

We want to extend a special acknowledgment to Elias Sandmann, who notified us of an error in the Transformer Lens documentation that affected our implementation of Logit Lens. The correction was crucial for the accuracy and reproducibility of our results which may have gone overlooked without his concern.

References

[1] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[3] Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. An overview of catastrophic ai risks. *arXiv preprint arXiv:2306.12001*, 2023.

[4] Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, Gillian Hadfield, et al. Managing ai risks in an era of rapid progress. *arXiv preprint arXiv:2310.17688*, 2023.

[5] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

[6] Cody Rushing and Neel Nanda. Explorations of self-repair in language models. *arXiv preprint arXiv:2402.15390*, 2024.

[7] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*, 2022.

[8] Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024.

[9] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

[10] 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. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[11] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. *arXiv preprint arXiv:2312.09230*, 2023.

[12] Callum McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding an attention head. *arXiv preprint arXiv:2310.04625*, 2023.

[13] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

[14] Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

[15] Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*, 2023.

[16] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[17] Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pages 27011–27033. PMLR, 2023.

[18] Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*, 2023.

[19] Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, et al. Deja vu: Contextual sparsity for efficient llms at inference time. In *International Conference on Machine Learning*, pages 22137–22176. PMLR, 2023.

[20] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

[21] Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

[22] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[23] Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah. High-low frequency detectors. *Distill*, 2021. doi: 10.23915/distill.00024.005. https://distill.pub/2020/circuits/frequency-edges.

[24] Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. Curve circuits. *Distill*, 2021. doi: 10.23915/distill.00024.006. https://distill.pub/2020/circuits/curve-circuits.

[25] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

[26] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10231–10241, 2021.

[27] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Circuit component reuse across tasks in transformer language models. *arXiv preprint arXiv:2310.08744*, 2023.

[28] Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? *arXiv preprint arXiv:2310.17191*, 2023.

[29] Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*, 2023.

[30] Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

[31] Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*, 2023.

[32] Alexandre Variengien and Eric Winsor. Look before you leap: A universal emergent decomposition of retrieval tasks in language models, 2023.

[33] Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *arXiv preprint arXiv:1612.07771*, 2016.

[34] Stanisław Jastrzębski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. *arXiv preprint arXiv:1710.04773*, 2017.

[35] Nostalgebraist. Interpreting gpt: The logit lens. `https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`, 2020.

[36] Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

[37] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

[38] Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*, 2024.

[39] Ofir Press, Noah A Smith, and Omer Levy. Improving transformer models by reordering their sublayers. *arXiv preprint arXiv:1911.03864*, 2019.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[41] Minjia Zhang and Yuxiong He. Accelerating training of transformer-based language models with progressive layer dropping. *Advances in Neural Information Processing Systems*, 33: 14011–14023, 2020.

[42] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

[43] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, 2023.

[44] Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. Analyzing redundancy in pretrained transformer models. *arXiv preprint arXiv:2004.04010*, 2020.

[45] Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. *arXiv preprint arXiv:2212.09095*, 2022.

[46] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[48] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[49] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[50] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[51] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[52] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[53] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[54] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[55] 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. *arXiv preprint arXiv:1803.05457*, 2018.

[56] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[57] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[58] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

[59] Anand Subramanian. torch_cka, 2021. URL `https://github.com/AntixK/PyTorch-Model-Compare`.

[60] Gonçalo M Correia, Vlad Niculae, and André FT Martins. Adaptively sparse transformers. *arXiv preprint arXiv:1909.00015*, 2019.

[61] Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. *arXiv preprint arXiv:2403.00824*, 2024.

[62] Neel Nanda, Senthooran Rajamanoharan, Janos Kramar, and Rohin Shah. Fact finding: Attempting to reverse-engineer factual recall on the neuron level, Dec 2023. URL `https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall`.

[63] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*, 2023.

[64] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*, 2023.

[65] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

[66] Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

[67] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*, 2018.

[68] Zhu Liu, Cunliang Kong, Ying Liu, and Maosong Sun. Fantastic semantics and where to find them: Investigating which layers of generative llms reflect lexical semantics. *arXiv preprint arXiv:2403.01509*, 2024.

[69] Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022.

[70] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29, 2016.

[71] Neel Nanda. Transformerlens, 2022. URL `https://github.com/neelnanda-io/TransformerLens`.

[72] Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective, 2023.

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: As stated in the abstract, we perform layerwise interventions to investigate the robustness of layers. These interventions are suggestive of phases of inference that form a hypothesis. The remainder of the paper is series of experiments that support various findings of this hypothesis, across three different model families. We emphasize that we are proposing a hypothesis that are result of performing experiments that suggest this hypothesis.

   Guidelines:

   - The answer NA 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 NA 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 provide a limitation section where we discuss how we did not investigate the true cause of model-to-model difference and the potential loss of important findings that are a result of excessive aggregation over tokens.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: We do not have theoretical assumptions and proof in the paper.

   Guidelines:

   - The answer NA 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: First we describe in detail what encompasses a layerwise swap and deletion, including a diagram and the components involved. We include the models that we use and how the weights were preprocessed for experimentation. We also describe the specific metrics we aggregate over (KL divergence, entropy, loss, etc) For our supporting experiments, we describe what component we measured the output of (MLP, Attention), and what calculation was performed (Norm, ratio, etc). For the logit lens technique and the measurement of predictive and suppressive neurons, we describe the multiplications of specific matrices to recreate the results and reference appropriate papers that provide a deeper analysis of the method.

   Guidelines:

   - The answer NA 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 provide a link to a Github which contains all of the code to recreate the experiments, currently anonymous. We also describe which specific model families and datasets (the Pile) were used the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We do not perform any training but we specify all the metrics we collect from the models, including KL divergence, loss, entropy, and logits. We also describe the preprocessing of the weights carried out before experimentation, further in the appendix.

   Guidelines:

   - The answer NA 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: All our plots contain error bands. On some plots the results of aggregation causes the plot to have fine error bars, often not visible. For specificity: Figure 1 does not have error bars, however, every peak in this figure appears with error bars in later sections of the paper. Figure 3 we display metrics over all the models without aggregating in place of showing an average line. Figures 4, 5, 6, 7, 8, 13 all contain error bands which are 1-sigma.

Guidelines:

- The answer NA 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 provide an appendix section titled “Additional Empirical Details" described compute usage.

   Guidelines:

   - The answer NA 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: We have read and confirm that we abide to the code of ethics.

   Guidelines:

   - The answer NA 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: At the beginning of this paper we discuss the importance of connecting mechanistic interpretability to the broader understanding of machine learning. We discuss how this work aims to make ML systems more understandable and allow for the detection of risks and vulnerabilities that can arise are language models scale. There are no negative impacts of the work as we write in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: None of our analysis, experiments, or conclusions pose a risk.

    Guidelines:

    - The answer NA 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: We utilize 3 different model families and a single dataset, the Pile, which are cited accordingly in the paper.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We do not use human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: There are no study participants.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We simply study LLMs, and do not rely on LLM usage for hypothesis or experiment creation or execution.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Experiment Diagram

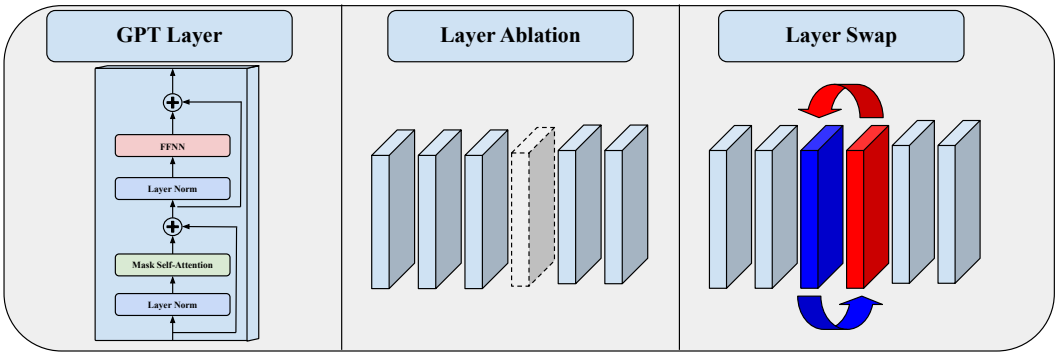


Figure 13: To study the stages of inference, we perform two experiments, each a layer-wise intervention, where a layer (left) encompasses all model components. The first intervention is a zero ablation (i.e, layer removal) of the layer (middle), in which a layer is fully removed and residual connections skip the layer entirely. The second intervention (last) is an adjacent layer swap, in which we permute the positions of two layers. The ablation is performed on all layers, while the layer swap is performed on all adjacent pairs of layers in the model.

Name	HuggingFace Model Name
Pythia 410M	EleutherAI/pythia-410m-deduped
Pythia 1.4B	EleutherAI/pythia-1.4b-deduped
Pythia 2.8B	EleutherAI/pythia-2.8b-deduped
Pythia 6.9B	EleutherAI/pythia-6.9b-deduped
Pythia 12B	EleutherAI/pythia-12b-deduped
GPT-2 Small (124M)	gpt2
GPT-2 Medium (355M)	gpt2-medium
GPT-2 Large (774M)	gpt2-large
GPT-2 XL (1.5B)	gpt2-xl
Phi 1 (1.3B)	microsoft/Phi-1
Phi 1.5 (1.3B)	microsoft/Phi-1.5
Phi 2 (2.7B)	microsoft/Phi-2
Qwen 0.5B	Qwen/Qwen2.5-0.5B
Qwen 1.5B	Qwen/Qwen2.5-1.5B
Qwen 3B	Qwen/Qwen2.5-3B
Qwen 14B	Qwen/Qwen2.5-14B
Llama-3.2 1B	meta-llama/Llama-3.2-1B
Llama-3.2 3B	meta-llama/Llama-3.2-3B
The Pile	EleutherAI/the_pile_deduplicated

Table 3: List of models and dataset used in the experiments.

B Additional Empirical Details

Github All experimental code for future experiments is available at: https://github.com/vdlad/Remarkable-Robustness-of-LLMs.

Transformer Lens We make ubiquitous use of Transformer Lens [71] to perform hooks and transformer manipulations.

HuggingFace For specificity, we utilize the following HuggingFace model names, and dataset. We do not change the parameters of the models from what they are described on the HuggingFace page.

All experiments described can be performed on a single NVIDIA A6000. We utilized 2 NVIDIA A6000 and 500 GB of RAM. To aggregate the metrics described in the paper, we run the model on 1

million tokens ℓ times, where ℓ is the number of layers. This takes on average 8 hours per model, per layer intervention (swapping and ablating). We save this aggregation for data analysis.

Residual Sharpening to Residual Calibration We initially named the final stage of inference "residual sharpening" but have renamed it to "residual calibration" for greater accuracy. While suppression neurons do eliminate obsolete features from the model's representation, the final layers sometimes exhibit an increase in entropy—a seemingly contradictory behavior if the goal were simply to sharpen predictions toward a single top token. Instead, this entropy increase suggests that suppression neurons calibrate the representation to produce a well-formed distribution over possible next tokens, aligning with the language modeling objective. This calibration process differs from sharpening, which would imply converging toward a single prediction. Additionally, models even within the same family exhibit varying entropy patterns in their final layers. We hypothesize that the variation in entropy of the final layers may indicate model confidence and contribute to hallucination behavior, though we leave this investigation to future work.

Layer Norm Preprocessing We utilize several conventional weight preprocessing techniques to streamline our calculations [71].

Following [14], before each MLP calculation, a layer norm operation is applied to the residual stream. This normalizes the input before the MLP. The TransformerLens package simplifies this process by incorporating the layer norm into the weights and biases of the MLP, resulting in matrices W_{eff} and b_{eff}. In many layer norm implementations, trainable parameters $\boldsymbol{\gamma} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^n$ are included:

$$\text{LayerNorm}(\mathbf{x}) = \frac{\mathbf{x} - \mathbb{E}(\mathbf{x})}{\sqrt{\text{Var}(\mathbf{x})}} * \boldsymbol{\gamma} + \mathbf{b}. \tag{1}$$

We "fold" the layer norm parameters into W_{in} by treating the layer norm as a linear layer and then merging the subsequent layers:

$$\mathbf{W}_{\text{eff}} = \mathbf{W}_{\text{in}}\ \mathbf{diag}(\boldsymbol{\gamma}) \qquad \mathbf{b}_{\text{eff}} = \mathbf{b}_{\text{in}} + \mathbf{W}_{\text{in}}\mathbf{b} \tag{2}$$

Additionally, we then center reading weights. Thus, we adjust the weights $\mathbf{W}_{\text{eff}}$ as follows:

$$\mathbf{W}_{\text{eff}}^{'}(i,:) = \mathbf{W}_{\text{eff}}(i,:) - \bar{\mathbf{W}}_{\text{eff}}(i,:)$$

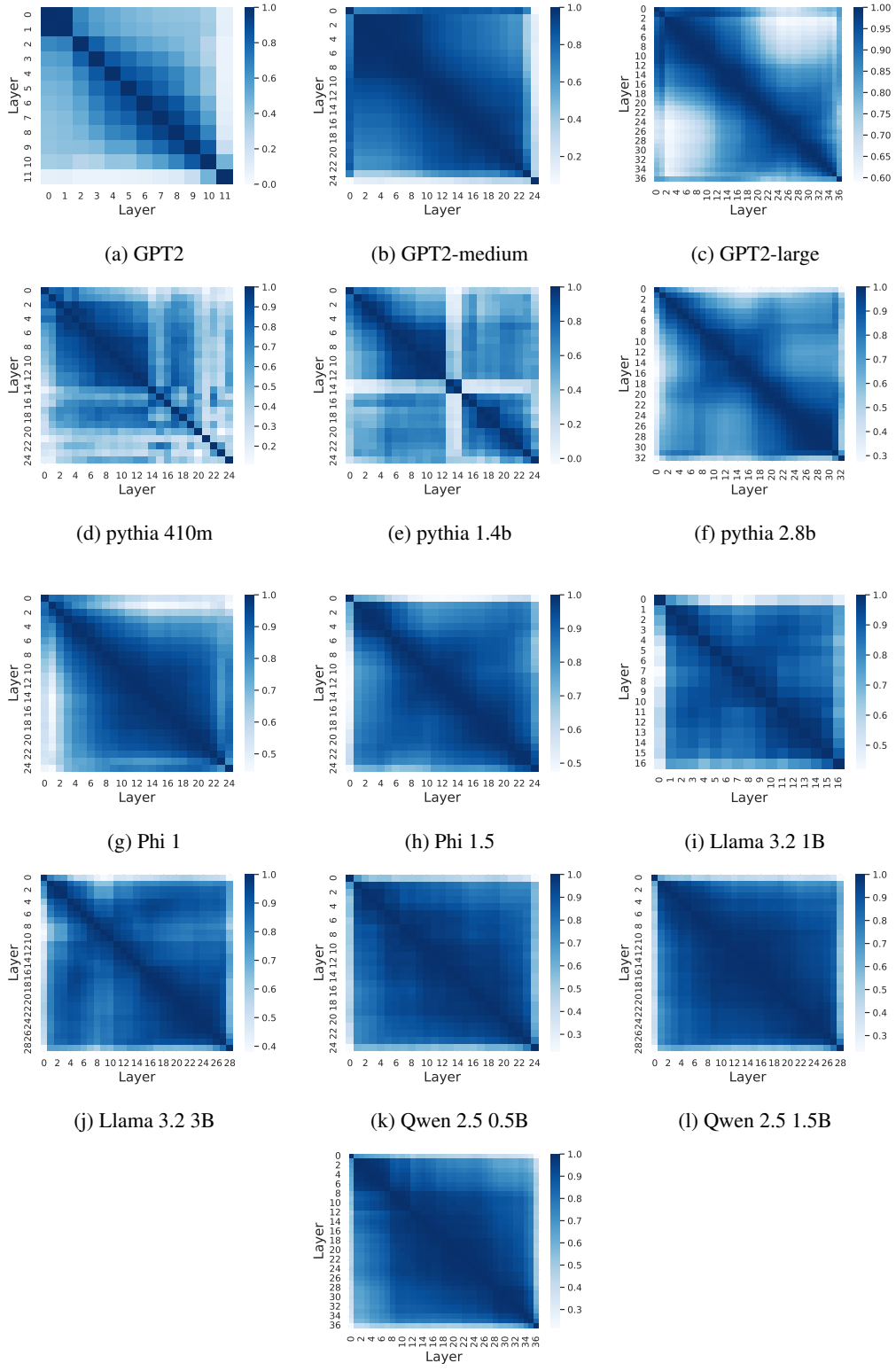
Centering Writing Weights Because of the LayerNorm operation in every layer, we can align weights with the all-one direction in the residual stream as they do not influence the model's calculations. Therefore, we mean-center $\mathbf{W}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ by subtracting the column means of $\mathbf{W}_{\text{out}}$:

$$\mathbf{W}_{\text{out}}^{'}(:,i) = \mathbf{W}_{\text{out}}(:,i) - \bar{\mathbf{W}}_{\text{out}}(:,i)$$

Extension of Results in Larger Models (>10B parameters) In Appendix K, we extend the results of the core experiments in larger models with more than 10B parameters (Qwen2.5-14B, Pythia-12B).

Societal Impact We do not anticipate any immediate societal impact from this research.

C Centered Kernel Alignment (CKA)



(a) GPT2 (b) GPT2-medium (c) GPT2-large

(d) pythia 410m (e) pythia 1.4b (f) pythia 2.8b

(g) Phi 1 (h) Phi 1.5 (i) Llama 3.2 1B

(j) Llama 3.2 3B (k) Qwen 2.5 0.5B (l) Qwen 2.5 1.5B

(m) Qwen 2.5 3B

Figure 14: CKA across layers from the last token representation sampled from Pile dataset (max token length 512, batch size 128). We used unbiased CKA [72, 59].

D Benchmark Tasks Performance After Layer-Wise Intervention

We evaluate the benchmark performance on HellaSwag, ARC-Easy and LAMBADA [54–56] with the intervened model. We observe a similar trend to KL divergence reported in the main paper. Generally, the intervention at the first layer and the last layer shows catastrophic deterioration of the performance but intervention on intermediate layers shows robust performance.

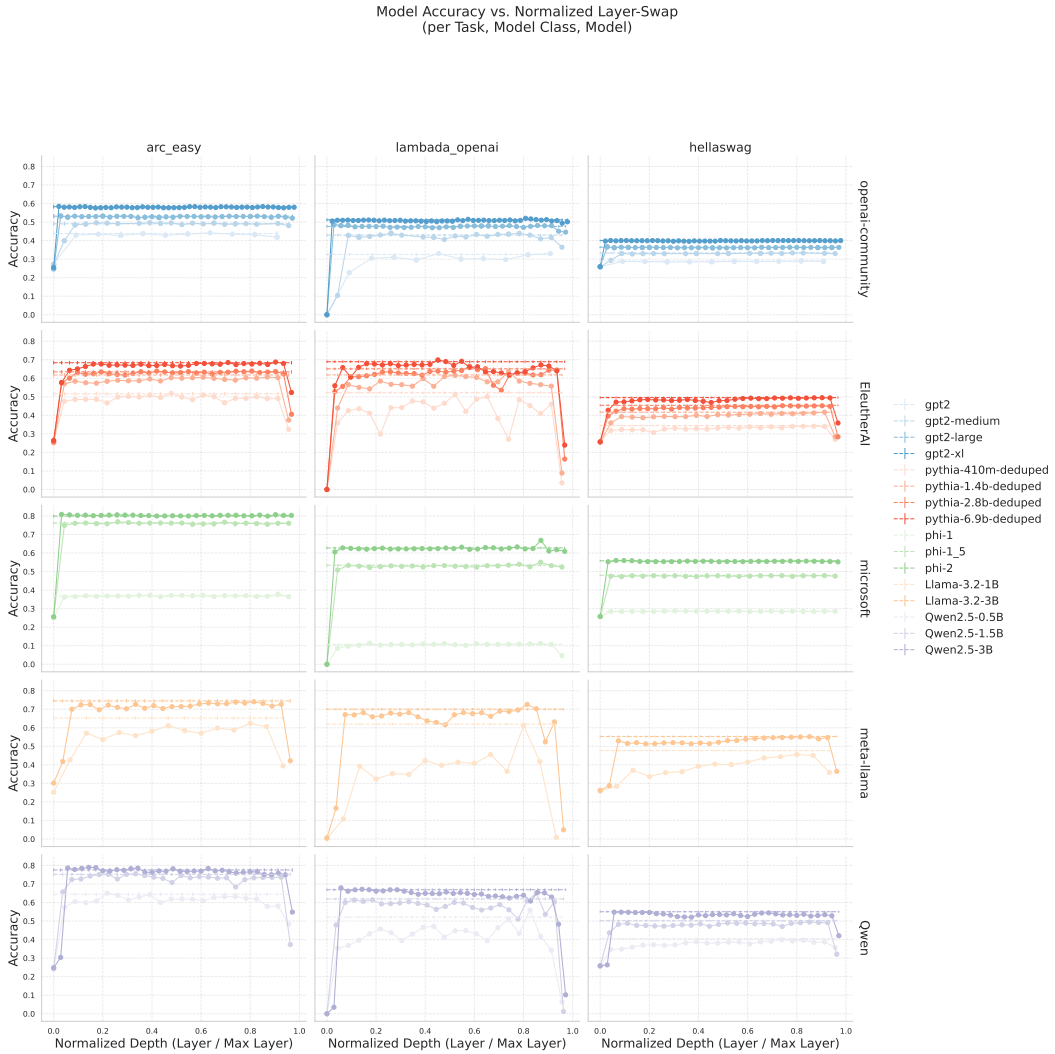


Figure 15: Benchmark task performance after layer swap. Baseline performance of each model is marked with a dotted horizontal line.

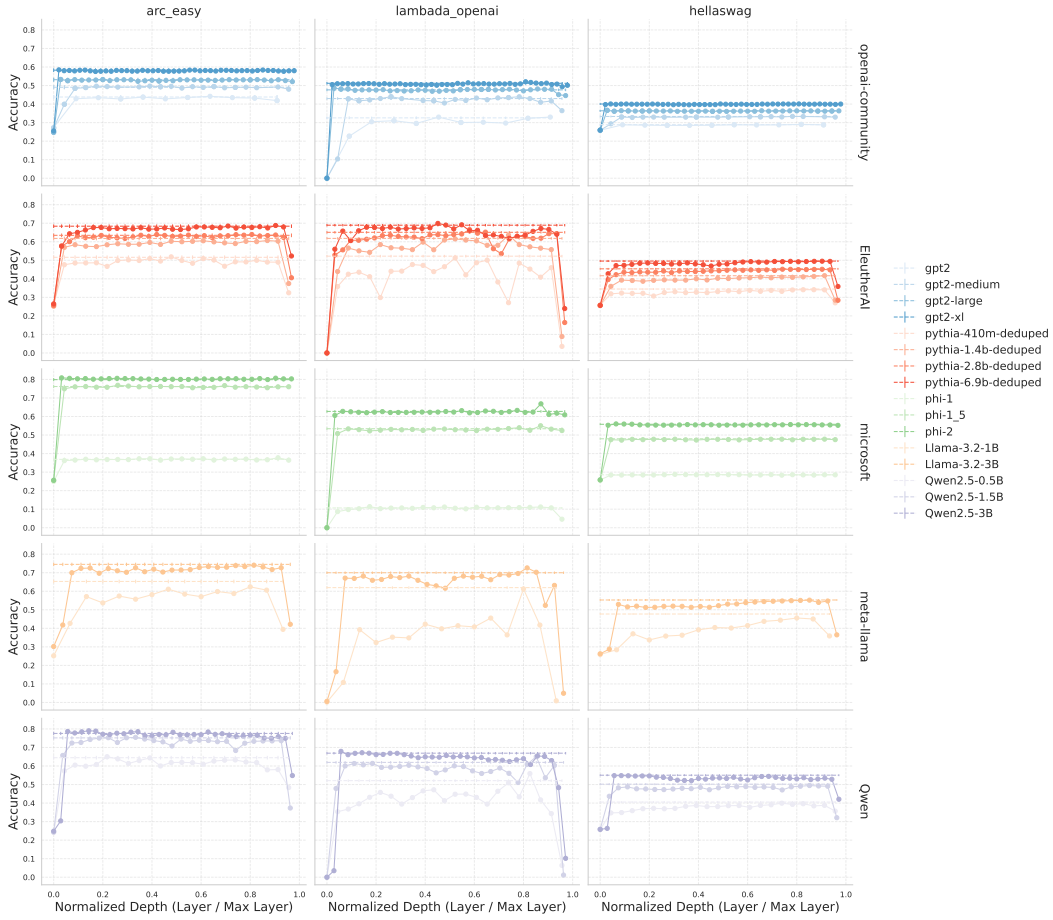
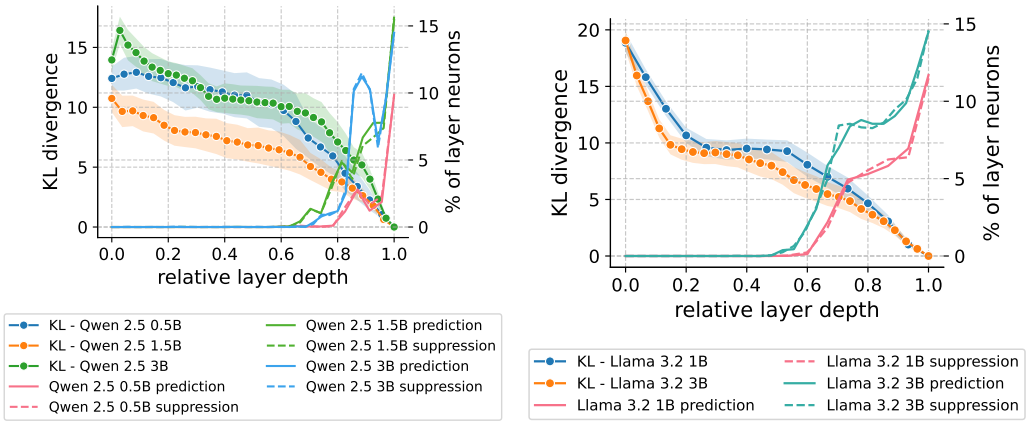


Figure 16: Benchmark task performance after layer swap. Baseline performance of each model is marked with a dotted horizontal line.

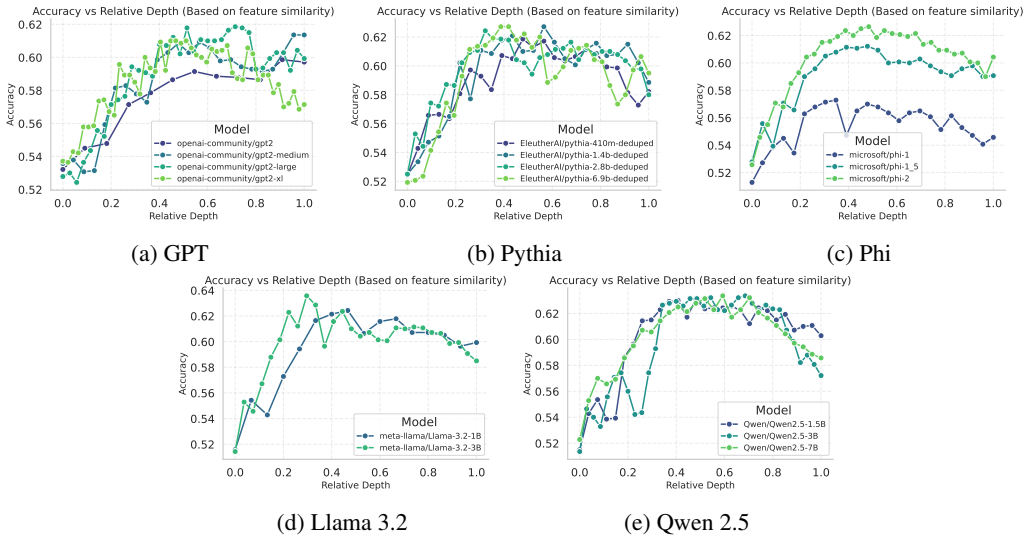
E Prediction and Suppression Neuron



(a) Qwen (b) Llama

Figure 17: Prediction and Suppression neurons for Qwen and Pythia.

F WiC contextual word probe

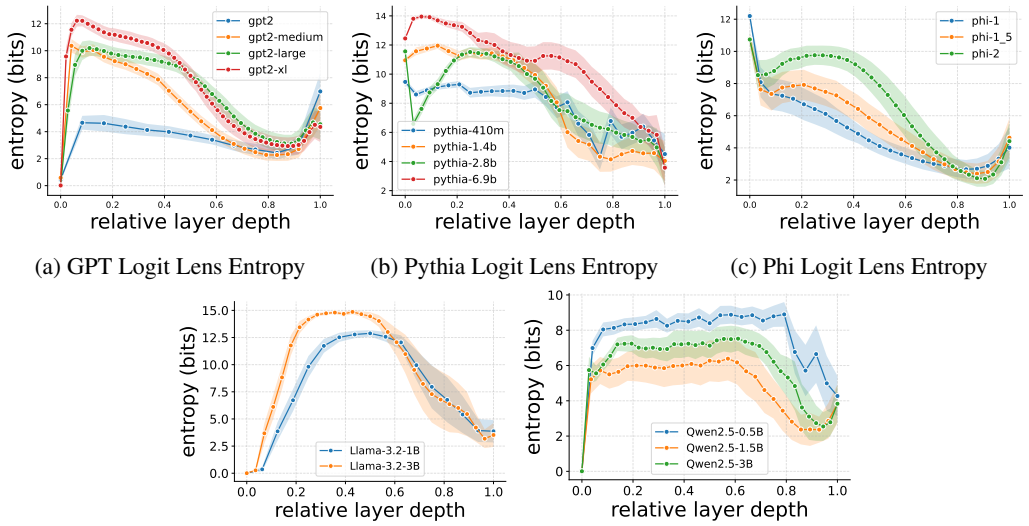


(a) GPT (b) Pythia (c) Phi

(d) Llama 3.2 (e) Qwen 2.5

Figure 18: WiC probing accuracy over layers across model families and sizes. Across all models and sizes, we observe the probe accuracy related to contextual semantics of lexical items gradually increases and peaks around the middle layers and degrades.

G Logit Lens Entropy

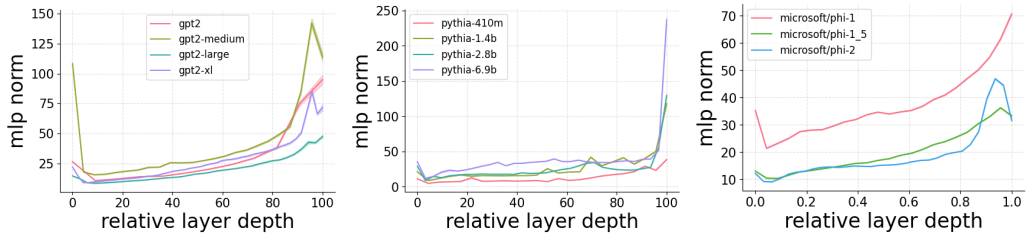


(a) GPT Logit Lens Entropy (b) Pythia Logit Lens Entropy (c) Phi Logit Lens Entropy

(d) Llama Logit Lens Entropy (e) Qwen Logit Lens Entropy

Figure 19: Using the logit lens technique [35], we calculate the probability distribution of the next token at the end of every layer, and then take its entropy.

H MLP Norms



(a) GPT MLP Output (b) Pythia MLP Output (c) Phi MLP Output

(d) Llama MLP Output (e) Qwen MLP Output

Figure 20: The norm of the output of every MLP across its layers to measure its contribution to the residual stream. Across all 18 models, the norm grows and peaks in the final layers before output, suggestive of the final two stages of inference, predictive ensembling, and residual calibration

I Top Prediction and Suppression Neurons

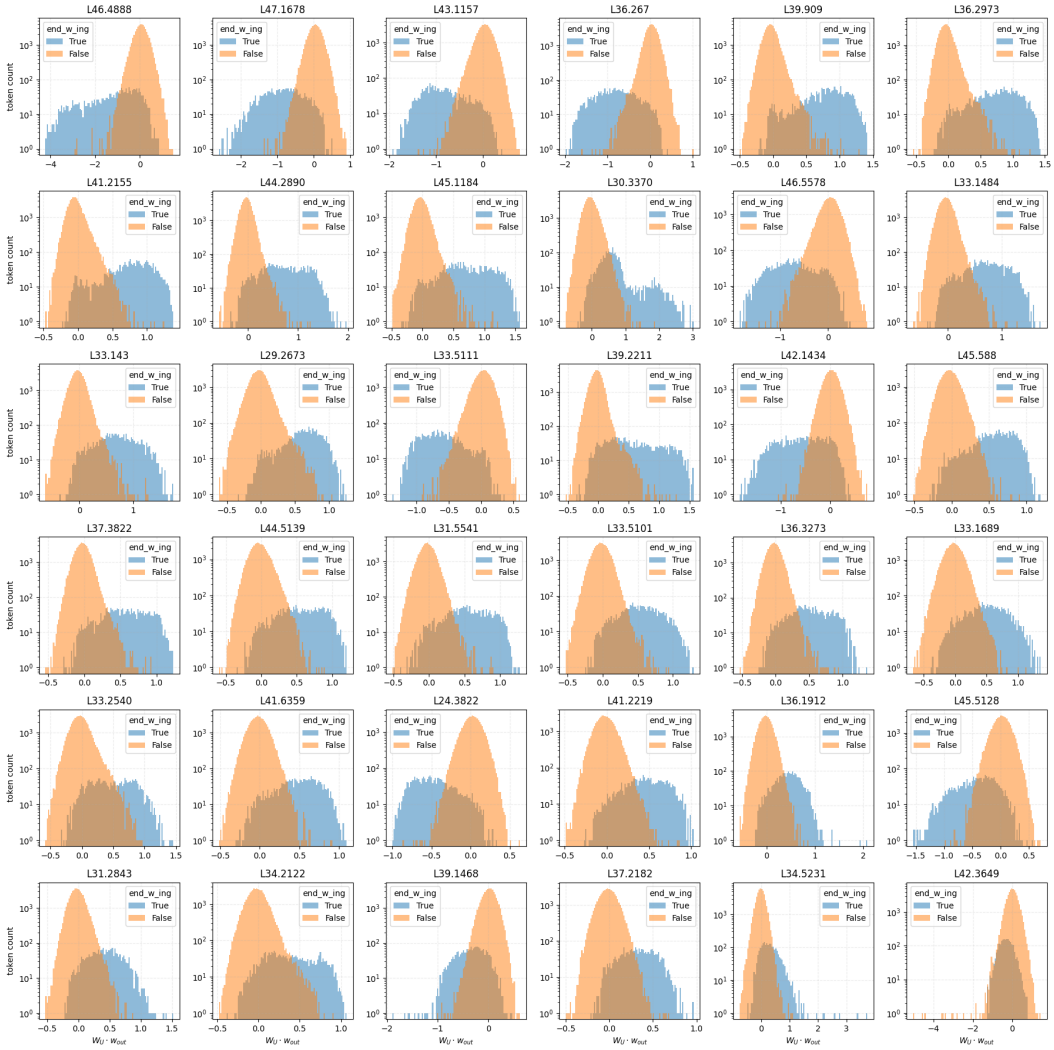
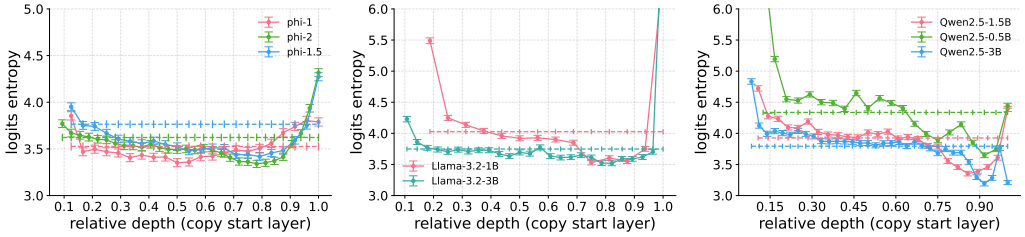
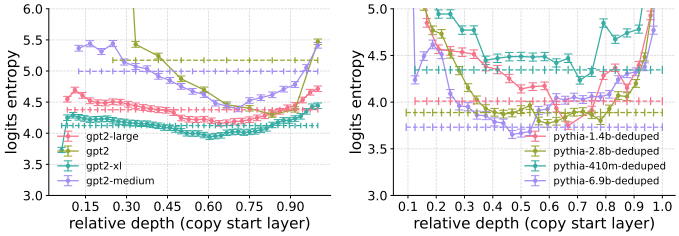


Figure 21: Top 36 prediction and suppression neurons for -ing which have the greatest mean absolute difference between respective ($W_U \cdot w_{\text{out}}$). Elements with a negative skew are suppression neurons for the respective labeled class, while elements with a positive skew are prediction neurons. This is calculated by calculating the product between the model unembedding weights and output weights of MLP.

J Layer repeats experiment

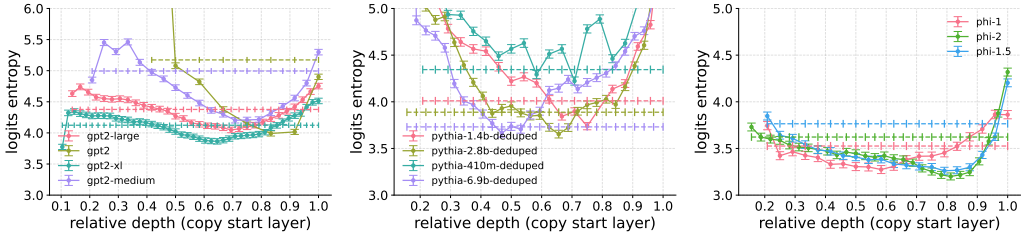


(a) Phi Block 3 (b) Llama 3.2 Block 3 (c) Qwen 2.5 Block 3

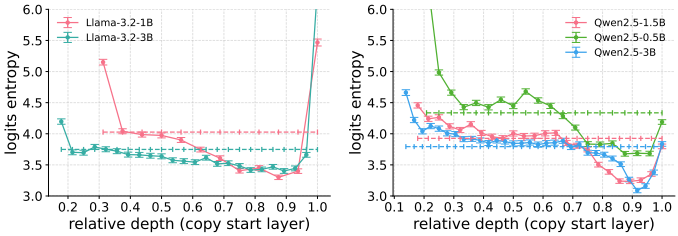


(d) GPT Block 3 (e) Pythia Block 3

Figure 22: Block 3 repeat experiment.



(a) GPT (b) Pythia (c) Phi Repeat Block 5



(d) Llama Repeat Block 5 (e) Qwen Repeat Block 5

Figure 23: Block 5 repeat experiment on additional models.

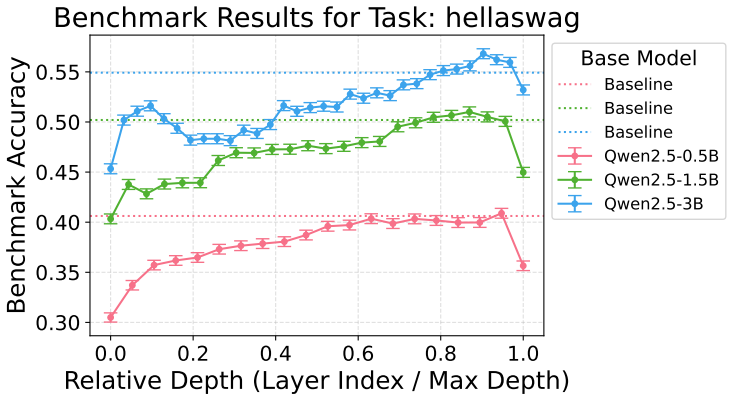
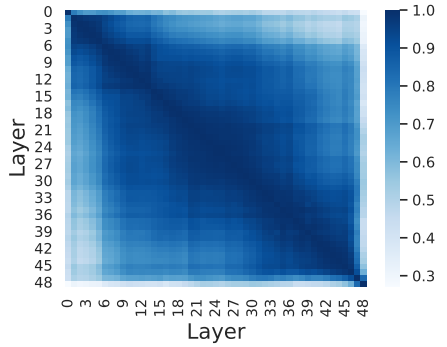
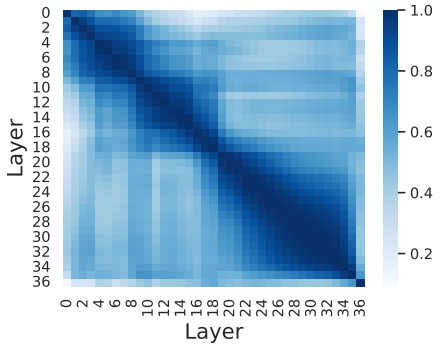


Figure 24: Qwen repeat 5 model's performance on Hellaswag

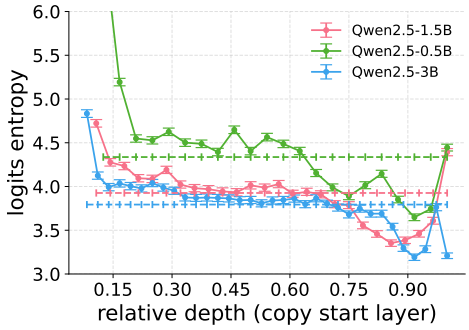
K Experiments on Larger Models (>10B parameters)



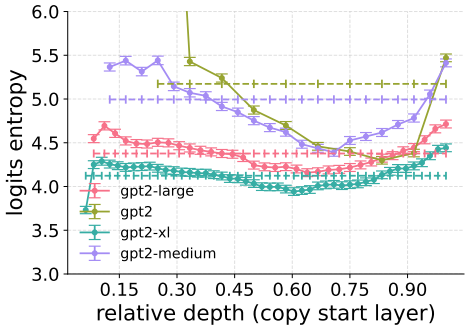
(a) CKA, Qwen2.5-14B



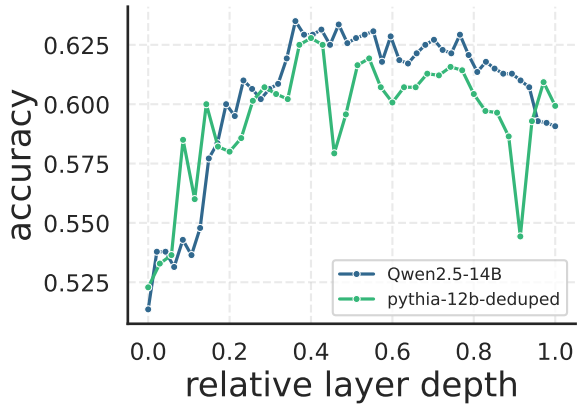
(b) CKA, Pythia-12B



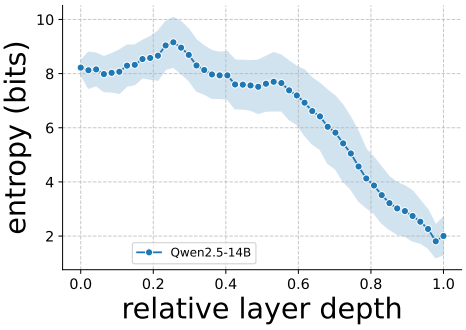
(c) Logit Entropy, Qwen2.5-14B



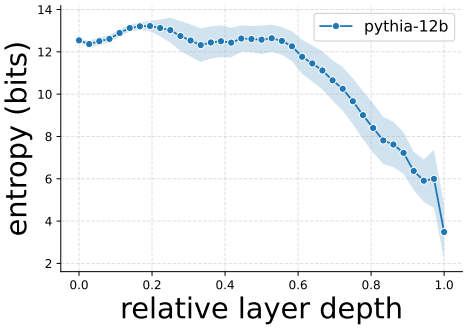
(d) Logit Entropy, Pythia-12B



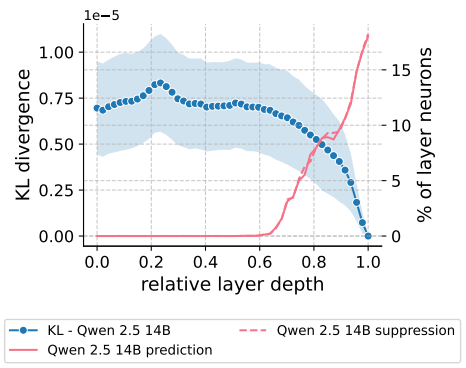
(e) WiC Task



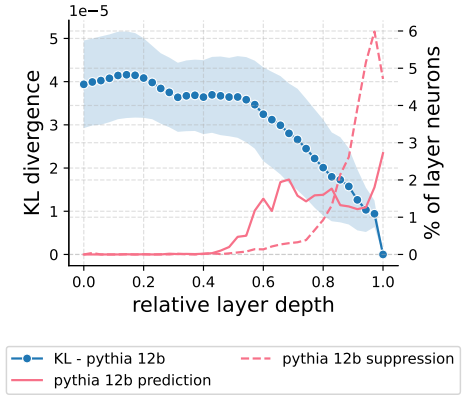
(f) Logit Lens Entropy, Qwen2.5-14B



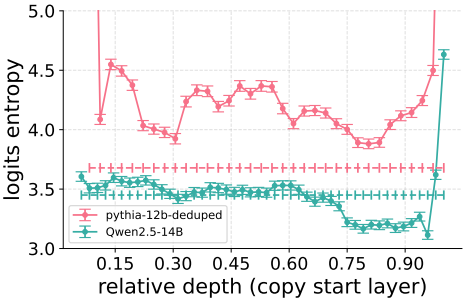
(g) Logit Lens Entropy, Pythia-12B



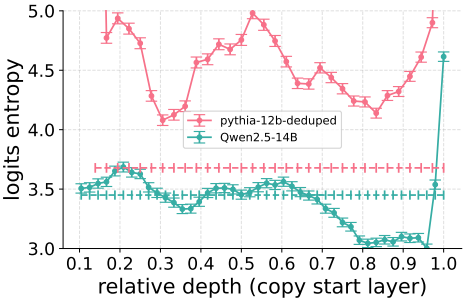
(h) Prediction/Suppression Neurons, Qwen2.5-14B



(i) Logit Lens Entropy, Pythia-12B



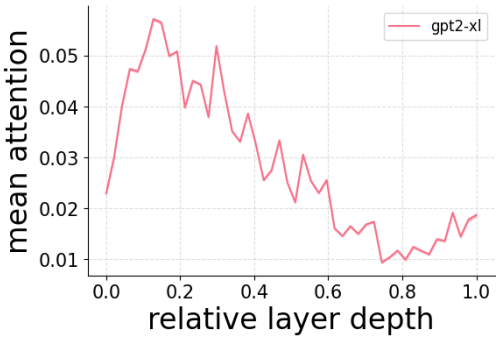
(j) Block 3 repeat experiments



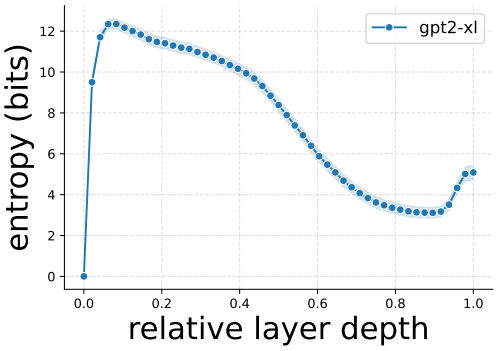
(k) Block 5 repeat experiments

Figure 25: Extension of Experiments on Larger Models (>10B).

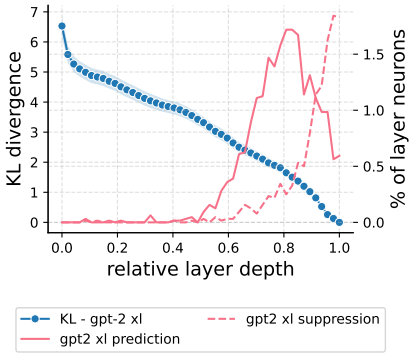
L Experiment Subset on OOD Data



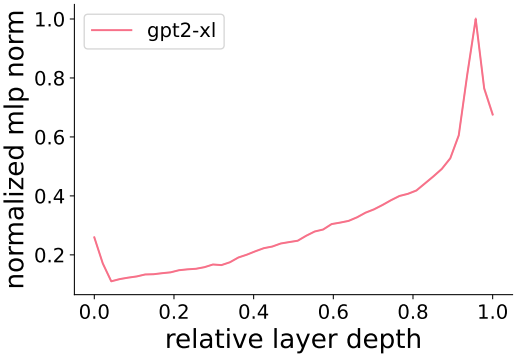
(a) Mean Attention, recent arXiv papers



(b) Logit Lens Entropy, recent arXiv papers



(c) KL divergence between intermediate and output distributions, recent arXiv papers



(d) MLP Output Norms, recent arXiv papers

Figure 26: Running on 2024–2025 arXiv papers, code, and languages results in consistent patterns across the hypothesized stages of inference for GPT-XL.