FROM UNITS TO LAYERS: STRUCTURAL INTERVENTION FOR LLM HALLUCINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are known to exhibit hallucination, producing fluent but factually incorrect outputs. Existing approaches largely treat hallucination as a black-box phenomenon, offering limited internal insight. Prior mechanistic studies of hallucinations have focused on individual computational units, whose ablation tends to suppress the targeted behavior at the cost of performance on unrelated tasks. We argue that this cost is structural. Individual units may participate in many functions at once, which removal almost necessarily perturbs unrelated computation. In this work, we propose a structure-level alternative. We localize hallucination-related units via attribution and causal validation, and map them onto the model's layer-wise structure. It reveals that hallucination concentrates in specific structural regions that are stable across tasks. Building on this insight, we introduce a set of structure-level intervention operators, including *Freeze*, *Swap*, and *Insert*, which modify the layer-wise computation trajectory through block skipping, reordering, and relocation, respectively. To our knowledge, this is the first systematic class of structure-level operations for hallucination mitigation in LLMs. We systematically evaluate these operators against unit-level ablation in both intra-task and cross-task settings, spanning benchmarks for knowledge-based question answering, grammatical acceptability, and reasoning. Our solution achieves comparable to or stronger performance than the state-of-the-art, indicating that hallucination is governed by higher-level computational organization rather than by isolated units. It highlights structure-level intervention is a more principled basis for behavioral control in LLMs.

1 INTRODUCTION

Despite the broad capabilities of large language models (LLMs) across wide range of tasks, i.e., question answering, reasoning and code generation, hallucination remains one of the most persistent failure modes of LLMs Huang et al. (2025); Ji et al. (2023); Naveed et al. (2025). A substantial body of work has developed methods to detect or mitigate hallucination across the entire LLM lifecycle, from pre-training interventions to inference-time methods. Many methods operate on model outputs rather than on the computations that produce hallucination, resulting in the mitigation as necessarily reactive not mechanistic Zhang et al. (2025). A more systematic approach would be by intervening the internal computations for hallucinated outputs. However, it remains unclear at what level of granularity such interventions should operate.

Mechanistic interpretability (MI) provides a promising pathway by acting at the granularity of individual computational units Somvanshi et al. (2026). Existing studies have so far identified specific attention heads, MLP neurons, and residual stream Sharkey et al. (2025); Bereska & Gavves (2024); Méloux et al. (2025), and intervene by removing or modifying them Gao et al. (2025); Bashir et al. (2025). We refer to these collectively as *unit-level interventions*. While these works have been effective in suppressing specific behaviors in controlled settings, a recurring observation is that it often comes at a cost on unrelated tasks, leading to systematic impacts on overall model capability rather than localized corrections Elhage (Dec 22, 2021); Wang et al. (2023a); Elhage et al. (2022). We argue that this may arise from the highly distributed and entangled nature of transformer representations, where multiple functionalities share overlapping computational pathways. Consequently, removing a "problematic" unit that participates in the targeted behavior almost inevitably perturbs unrelated computations that route through the same substrate.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In contrast, recent analyses suggest that the layer-wise organization carries functional structure that unit-level analyses do not capture. Different layers exhibit functional specialization and coordinated roles, and preserving this hierarchy is critical for stable information flow Geva et al. (2021); Sun et al. (2025). This motivates an alternative granularity. Instead of removing individual units, performing structure-level adjustments may offer a more robust trade-off between behavioral control and capability preservation. This raises the question of this paper: Can we design structure-level interventions that suppress undesirable behaviors more robustly than unit-level alternatives while preserving performance on unrelated tasks?

In this paper, we investigate the internal mechanisms of hallucination in LLMs from a mechanistic interpretability perspective, with a focus on their structural and cross-task behavior. We ask whether hallucination related computation is concentrated in identifiable structural regions of LLMs, and whether intervening at those regions yields a better trade-off between hallucination suppression and preservation of unrelated task capabilities. To do so, we first use attribution method to identify candidate units associated with hallucinated outputs across attention heads, MLP neurons and residual stream dimensions. We then project these signals onto the model's layer-wise structure to uncover their structural organization, moving beyond unit-level explanations toward a structured understanding of hallucination. This projection reveals that hallucination concentrates in specific structural regions that are stable across tasks, providing the empirical foundation for an alternative to intervention. Built on this basis, we design a series of structure-level operations that act on entire layer-wise regions, and that together form a coherent class of interventions that operate on complete Transformer blocks and modify their participation or position in the layer-wise computation trajectory. To our knowledge, this is the first systematic formulation of structure-level operations as an alternative class for behavioral control in LLMs.

We evaluate these operations across diverse task categories, including knowledge-based question answering Lin et al. (2022), grammatical acceptability Warstadt et al. (2019), and reasoning tasks Miao et al. (2020), in both intra-task and cross-task settings. The reason for selecting these tasks is to share no surface overlap with the hallucination signal in the tasks. The cross-task design is essential to the argument, since the structure vs units question is fundamentally about what get preserved when behavior is supprsssed, not only about what gets suppressed. Our results reveal that hallucination is not uniformly distributed across the model, but instead concentrates in specific structural regions that are stable across tasks. More importantly, compared to unit-level interventions, structure-level methods exhibit significantly stronger cross-task robustness: they effectively reduce hallucination while preserving, and in some cases improving, performance on other tasks. These findings suggest that hallucination is governed by higher-level computational structure rather than isolated local units, highlighting structure-level intervention as a more principled approach for behavioral intervention. Our main contributions are as follows:

1. We formulate the granularity question for behavioral intervention in LLMs and argue why structure-level interventions should offer a more robust trade-off than unit-level intervention.

2. We introduce a class of structure-level operations, that act on layer-wise regions rather than on individual units. To our knowledge, this is the first systematic alternative to unit-level ablation for behavioral control in LLMs.

3. We localize hallucination related computation, and show that its layer-wise projection reveals stable structural concentration rather than diffuse unit-level distribution. This empirical finding motivate our design of structure-level approach independently of the operations themselves.

4. Through extensive evaluation in both intra-task and cross-task settings, we demonstrate the substantially improved performance with structure-level operations. We also provide detailed case studies, including fitting-layer analysis and targeted pre-convergence interventions, to provide mechanistic insights into how hallucination emerges and how structural interventions influence model behavior.

2 Related Work

2.1 Hallucination in Large Language Models

Existing research on hallucination can be broadly divided into detection and mitigation approaches. Detection methods typically operate at the output level, using external knowledge bases, retrieval

systems Guu et al. (2020), or auxiliary models to verify factual consistency. For example, retrieval-augmented generation (RAG) grounds model outputs with external documents Lewis et al. (2020); Asai et al. (2024), while post-hoc verification assesses consistency with reference knowledge Lin et al. (2022); Manakul et al. (2023). Mitigation has been explored throughout the LLM lifecycle. At training time, supervised fine-tuning (SFT) Ouyang et al. (2022), reinforcement learning from human feedback (RLHF) Kaufmann et al. (2025), and data-centric methods Wang et al. (2023b); Zhou et al. (2023) aim to improve factual reliability. At inference time, decoding constraints Li et al. (2023), retrieval augmentation Izacard et al. (2023), and iterative refinement Wei et al. (2022); Madaan et al. (2023) guide models toward more reliable outputs. Despite their practical effectiveness, these methods mainly intervene at the input–output level and provide limited insight into how hallucinations arise and propagate within the model.

2.2 Mechanistic Interpretability

Mechanistic interpretability (MI) seeks to understand neural network behavior by analyzing internal computational structures, identifying units such as neurons, attention heads, and their interactions Belrose et al. (2025). In Transformers, computation is organized as layer-wise residual updates, where attention modules route information and MLPs transform representations, leading to a hierarchical refinement of features Tenney et al. (2019). Existing MI methods Syed et al. (2024); Makelov et al. (2024); Li & Subramani (2026); Wang et al. (2023a); Wu et al. (2023); Gao et al. (2025) typically focus on attributing behaviors to individual units using techniques such as gradient-based attribution, activation patching, and circuit analysis. While effective, these approaches operate at a fine-grained level and often overlook the distributed nature of representations, where multiple functions share overlapping computation pathways. As a result, unit-level interventions can induce non-local effects and fail to capture how behaviors are organized across layers. In this work, we adopt a structural perspective, mapping hallucination behavior onto the hierarchical computation of Transformers, and investigate structure-level interventions (e.g., insert, swap, and freeze) to better control model behavior while preserving global computation during different benchmark.

3 Preliminaries

Transformer models can be formulated as compositions of layer-wise functions, $f(x) = F_L \circ F_{L-1} \circ \cdots \circ F_1(x)$, where each F_ℓ is a residual block containing attention and feed-forward modules Vaswani et al. (2017); Elhage (Dec 22, 2021). Representations are propagated through a shared residual stream, $r_{\ell+1} = r_\ell + \mathrm{Attn}_\ell(r_\ell) + \mathrm{MLP}_\ell(r_\ell)$, yielding a progressive refinement across layers Tenney et al. (2019); Belrose et al. (2025); Simoulin & Crabbé (2021). This computation relies on distributed representations and superposition: semantic features are typically encoded across multiple directions, while individual neurons or attention heads may participate in multiple functions Elhage et al. (2022). Consequently, model behavior emerges from coordinated computational pathways rather than isolated units. Such distributed and redundant structure implies that local interventions on individual components can induce non-local effects on the model's computation Bricken (2023).

Mechanistic interpretability often relies on **unit-level interventions**, such as ablating neurons Gao et al. (2025), attention heads Voita et al. (2019); Nam et al. (2025), or edges Bashir et al. (2025). However, such fine-grained interventions can have non-local effects because Transformer representations are distributed across overlapping computational pathways. We therefore consider **structure-level interventions** that operate on complete Transformer blocks within $f(x) = F_L \circ \cdots \circ F_1(x)$. Specifically, we study three modifications of the layer-wise computation trajectory: (1) *Freeze*, which skips a selected block; (2) *Swap*, which exchanges the positions of two blocks; and (3) *Insert*, which relocates an auxiliary block to the target position and skips its original execution. These correspond to block removal, reordering, and relocation, respectively (Appendix E for more details).

4 Methodology

Our method aims to identify hallucination-related units in large language models and analyze their behavior under structure-level interventions. As illustrated in Figure 1, the overall framework consists of two main parts: **(1) Hallucination unit identification.** We first construct a three-stage pipeline to localize and refine hallucination-related units: (i) attribution-based candidate localization,

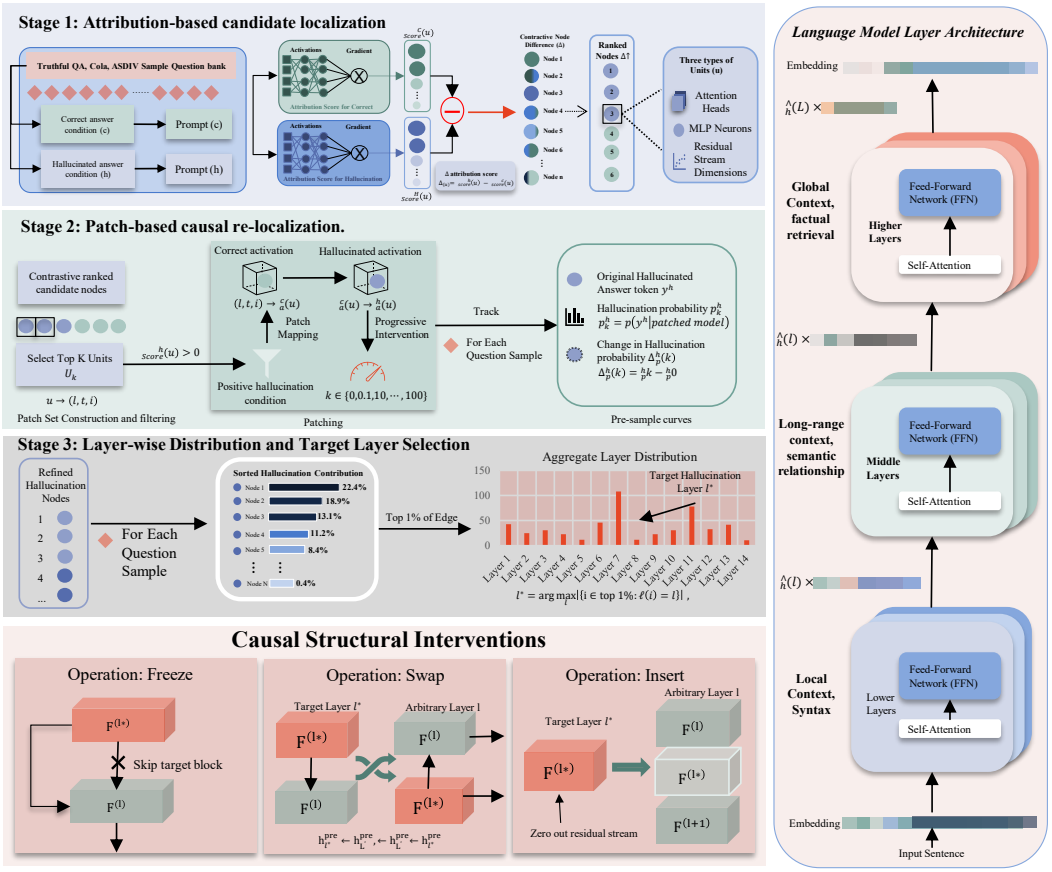


Figure 1: Methodology Overview

(ii) patch-based causal re-localization, and (iii) layer-wise aggregation. This process produces a structural distribution of hallucination-related units across model layers. **(2) Causal structural interventions.** Based on the identified layer distribution, we perform structure-level interventions, including *freeze*, *swap*, and *insert*, to modify the model's internal computation and evaluate their impact on hallucination and cross-task performance.

4.1 Identifying H-units via Attribution, Causal Refinement, and Layer Aggregation

Stage 1: Attribution-based candidate localization. Stage 1 is designed as an offline mechanistic localization procedure. We first localize a high-recall set of candidate units by contrasting the model's internal behavior under controlled correct and hallucinated answer spaces. For each question, we construct two prompts using correct and hallucinated answer sets, respectively, and obtain the model-preferred answer token in each condition. These answer sets provide supervision for defining a controlled contrastive attribution objective. We then backpropagate from the selected answer logit and compute attribution scores for internal units to efficiently identify units whose contributions differ between correct and hallucinated behavior. We consider three types of units within each layer: attention heads, MLP neurons, and residual stream dimensions. For each unit u, we define its attribution score using the activation–gradient product:

$$\mathrm{Score}^s(u) = \mathrm{Act}^s(u) \cdot \mathrm{Grad}^s(u), \qquad s \in \{c, h\}, \tag{1}$$

where $\mathrm{Act}^s(u)$ and $\mathrm{Grad}^s(u)$ denote the activation and its gradient under condition s, respectively.

To isolate hallucination-related units, we compute a contrastive attribution score:

$$\Delta(u) = \mathrm{Score}^h(u) - \mathrm{Score}^c(u), \tag{2}$$

and rank all units by $|\Delta(u)|$. Units with larger contrastive scores are selected as hallucination unit candidates, as they exhibit stronger condition-dependent contributions to the model output. Importantly, the oracle supervision and backward gradients used in this stage serve only as diagnostic instruments for mechanism discovery: they are used to identify where correct and hallucinated behaviors differ internally, rather than as inputs required by the subsequent structural intervention. Further implementation details, including prompt construction, hooked representations, and unit-level aggregation, are provided in Appendix A.

Stage 2: Patch-based causal re-localization. To refine the candidate set, we perform causal intervention based on activation patching, where selected units are modified under the hallucination condition. Given the contrastive ranking from Stage 1, we select a subset of top-ranked units according to a proportion parameter k and construct a patch set. For each selected unit u, we replace its activation under the hallucination condition with its corresponding value from the correct condition:

$$a^h(u) \leftarrow a^c(u), \qquad p_k^h = p(y^h \mid \text{patched model}), \qquad \Delta p^h(k) = p_k^h - p_0^h. \tag{3}$$

In practice, we apply this intervention only to units with positive hallucination contribution, i.e., $\mathrm{Score}^h(u) > 0$, to focus on units that actively promote hallucinated outputs. To evaluate the causal effect, we progressively increase the patching proportion k and measure the resulting change in hallucination probability. Units that consistently reduce hallucination probability are retained as refined hallucination units. Stage 2 retains only approximately half of the Stage 1 candidates across models, confirming that attribution alone produces a substantially broader candidate set (additional details are provided in Appendix B).

Stage 3: Layer aggregation. We further analyze the structural distribution of refined hallucination units at the layer level. For each sample i, we select a subset of top-ranked refined units and group them by their layer index. Let $\mathcal{U}_i$ denote the selected unit set. We define the per-sample and aggregated layer-wise distributions as:

$$C_i(l) = \sum_{u \in \mathcal{U}_i} \mathbf{1}[\mathrm{layer}(u) = l], \qquad \bar{C}(l) = \frac{1}{N} \sum_{i=1}^{N} C_i(l), \tag{4}$$

where $C_i(l)$ counts the number of selected units assigned to layer l for sample i, and $\bar{C}(l)$ represents the average distribution across all N samples.

The resulting layer distribution characterizes where hallucination-related units are concentrated in the model hierarchy and serves as the basis for subsequent structural interventions. The complete cross-model layer-wise distributions are provided in Appendix J.1. Importantly, the target layer is selected solely from the Stage-3 aggregation of causally refined hallucination-related units, without using any downstream structural intervention outcomes. We further validate this *ex ante* localization criterion through matched random controls and an exhaustive layer-wise Freeze analysis in section 6.3. Layer aggregation implementation details are also provided in Appendix C.

4.2 Causal Structural Interventions

We study structure-level interventions that modify the participation or position of complete Transformer blocks along the layer-wise computation trajectory. Let $h^{(l)}$ denote the representation entering Transformer block l, and let $F^{(l)}(\cdot)$ denote the complete transformation implemented by that block. The standard forward computation is

$$h^{(l+1)} = F^{(l)}(h^{(l)}). \tag{5}$$

Based on the layer aggregation results from Stage 3, we identify a target layer l^*. For interventions involving a second block, we denote its index by a and refer to it as the auxiliary layer. We consider three representative modifications of the layer-wise computation trajectory: block removal, reordering, and relocation (additional details are provided in Appendix D).

Freeze (block skipping). Freeze replaces the target block $F^{(l^*)}$ with the identity mapping:

$$\tilde{h}^{(l^*+1)} = h^{(l^*)}. \tag{6}$$

All other blocks remain unchanged. Freeze therefore removes one complete block computation from the forward trajectory while preserving the representation entering that block.

Swap (block reordering). Given the target layer l^* and an auxiliary layer a, Swap exchanges the positions of the two complete Transformer blocks:

$$\tilde{F}^{(j)} = \begin{cases} F^{(a)}, & j = l^*, \\ F^{(l^*)}, & j = a, \\ F^{(j)}, & \text{otherwise.} \end{cases} \tag{7}$$

The parameters and internal computations of the two blocks remain unchanged; only their positions in the forward trajectory are exchanged. This intervention evaluates the sensitivity of model behavior to the ordering and structural location of block transformations.

Insert (block relocation and composition). Given target layer l^* and auxiliary layer a, Insert relocates $F^{(a)}$ to the target position and applies it immediately before $F^{(l^*)}$, the auxiliary block is executed exactly once, at the target position:

$$\tilde{h}^{(l^*+1)} = F^{(l^*)}\Big(F^{(a)}(\tilde{h}^{(l^*)})\Big), \qquad \tilde{h}^{(a+1)} = \tilde{h}^{(a)}. \tag{8}$$

5 Experimental Setup

5.1 Implementation Details

We evaluate our method on multiple autoregressive Transformer models, including GPT-2 (Small, Medium, Large), Pythia-2.8B, and LLaMA-3-8B, to ensure robustness across model scales and architectures. All experiments are conducted on a Dell Precision 7960 Tower Workstation. We consider three task categories: (1) hallucination evaluation using TruthfulQA-style multiple-choice question answering with controlled correct and hallucinated answer spacesLin et al. (2022), (2) language understanding using CoLA for grammatical acceptability classificationWarstadt et al. (2019), and (3) reasoning tasks using ASDIV benchmarks requiring multi-step reasoningMiao et al. (2020) (Appendix G for more details and Limitation statement on appendix I). All experiments are implemented using TransformerLens and we collect residual, attention, and MLP activations and compute attribution scores using the activation–gradient product. Structural interventions (freeze, swap, insert) are applied at the layer level by modifying the forward computation.

5.2 Evaluation Metrics and Baselines

We evaluate hallucination using the probability of the hallucinated answer token $p^h = p(y^h)$, and report the change after intervention Δp^h. For task performance, we use accuracy on CoLA and exact match on reasoning tasks. In addition to these metrics, we analyze the relationship between the layer-wise distribution of hallucination-related components and the effectiveness of patch-based interventions. To identify candidate components, we adopt an attribution-based approach inspired by attribution patching Nanda (2024); Syed et al. (2024); Hanna et al. (2024), which estimates the contribution of internal activations via activation-gradient interactions. We compare against the baseline model without intervention and unit-level ablation(U) based on attribution scores. Furthermore, we evaluate structure-level interventions, including freeze, swap, and insert. All methods are evaluated under the same intervention budget to ensure fair comparison.

Model	Task	B	U	F	S	I
GPT2-S	TQA	0.6467	0.4000 (↓0.2467)	0.4249 (↓0.2218)	**0.3274 (↓0.3193)**	0.4390 (↓0.2077)
	CoLA	0.6600	**0.9434 (↑0.2834)**	0.8362 (↑0.1762)	0.7853 (↑0.1253)	0.8237 (↑0.1637)
	ASDIV	0.2400	**0.6547 (↑0.4147)**	0.4886 (↑0.2486)	0.4570 (↑0.2170)	0.6260 (↑0.3860)
GPT2-M	TQA	0.5493	0.2821 (↓0.2672)	**0.1021 (↓0.4472)**	0.2145 (↓0.3348)	0.3345 (↓0.2148)
	CoLA	0.6697	**0.9769 (↑0.3072)**	0.9152 (↑0.2455)	0.9064 (↑0.2367)	0.9378 (↑0.2681)
	ASDIV	0.3403	0.4774 (↑0.1371)	**0.9293 (↑0.5890)**	0.7853 (↑0.4450)	0.6565 (↑0.3162)
GPT2-L	TQA	0.4902	0.3108 (↓0.1794)	**0.1314 (↓0.3588)**	0.2068 (↓0.2834)	0.3105 (↓0.1797)
	CoLA	0.6600	**0.9925 (↑0.3325)**	0.7694 (↑0.1094)	0.8306 (↑0.1706)	0.9082 (↑0.2482)
	ASDIV	0.3133	0.8436 (↑0.5303)	**0.8913 (↑0.5780)**	0.5615 (↑0.2482)	0.6292 (↑0.3159)
Pythia-2.8B	TQA	0.4001	0.0930 (↓0.3071)	**0.0838 (↓0.3163)**	0.1553 (↓0.2448)	0.0959 (↓0.3042)
	CoLA	0.6870	**0.8996 (↑0.2126)**	0.8943 (↑0.2073)	0.8240 (↑0.1370)	0.8775 (↑0.1905)
	ASDIV	0.2396	**0.8858 (↑0.6462)**	0.5076 (↑0.2680)	0.5085 (↑0.2689)	0.5448 (↑0.3052)
LLaMA3-8B	TQA	0.7624	0.1305 (↓0.6319)	0.3273 (↓0.4351)	**0.1063 (↓0.6561)**	0.1136 (↓0.6488)
	CoLA	0.7931	0.9179 (↑0.1248)	0.8896 (↑0.0965)	0.9689 (↑0.1758)	**0.9852 (↑0.1921)**
	ASDIV	0.7694	0.9111 (↑0.1417)	0.9232 (↑0.1538)	0.9376 (↑0.1682)	**0.9636 (↑0.1942)**

Table 1: Intra-task results across five models and three tasks (TQA, CoLA, ASDIV). Values outside parentheses denote final performance, while values inside parentheses indicate changes relative to the baseline. For TQA, lower is better; for CoLA and ASDIV, higher is better. Yellow and red cells highlight the best unit-level and structure-level results, respectively, while blue cells indicate the second-best overall result.

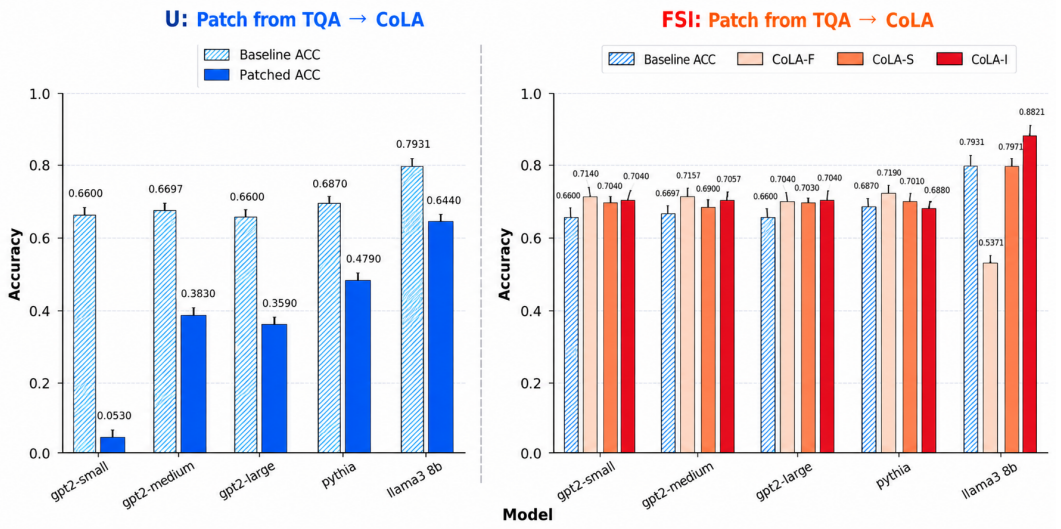


Figure 2: Cross-task transfer performance across benchmarks. Left: unit-level transfer from TQA to CoLA. Right: structure-level transfer from TQA to CoLA.

6 Result

6.1 Main Results

Hallucination Reduction. We first present the overall intra-task results in Table 1, where we evaluate five models across three benchmarks under four intervention strategies (U, F, S, I). For each benchmark, interventions are derived from its own localization results: Stage 1 attribution and Stage 2 causal refinement identify benchmark-specific units, with unit-level intervention (U) acting on the refined units and structure-level interventions (F, S, I) operating on the corresponding layer-wise regions obtained through Stage 3 aggregation. Different metrics are used across tasks: for TQA, we report hallucination-related probability (lower is better), while for CoLA and ASDIV, we follow the original task settings and report accuracy (higher is better). Overall, all models improve across the three benchmarks under the applied interventions. In particular, structure-level interventions (F, S, I) achieve comparable, and in many cases stronger, hallucina-

Table 2: Exhaustive single-layer Freeze analysis for Pythia-2.8B on TQA. The baseline hallucinated-answer probability is 0.4001. Positive reductions indicate decreased hallucination. **The Stage-3-selected Layer 10 is highlighted in bold.**

Layers 0–7			Layers 8–15			Layers 16–23			Layers 24–31		
Layer	**Prob.**	**Red.**	**Layer**	**Prob.**	**Red.**	**Layer**	**Prob.**	**Red.**	**Layer**	**Prob.**	**Red.**
0	0.6316	−0.2315	8	0.5390	−0.1389	16	0.2614	+0.1387	24	0.2651	+0.1350
1	0.4214	−0.0213	9	0.1281	+0.2721	17	0.2957	+0.1044	25	0.4862	−0.0861
2	0.4616	−0.0615	**10**	**0.0838**	+0.3163	18	0.4011	−0.0010	26	0.2168	+0.1833
3	0.2724	+0.1277	11	0.7451	−0.3450	19	0.4604	−0.0603	27	0.4367	−0.0366
4	0.2364	+0.1637	12	0.6892	−0.2891	20	0.3409	+0.0593	28	0.3940	+0.0061
5	0.4069	−0.0068	13	0.4568	−0.0567	21	0.2743	+0.1258	29	0.5544	−0.1543
6	0.4548	−0.0547	14	0.2125	+0.1876	22	0.5316	−0.1315	30	0.3464	+0.0537
7	0.2175	+0.1826	15	0.3679	+0.0322	23	0.5575	−0.1574	31	0.7254	−0.3253

tion reduction than unit-level ablation (U). This pattern holds across model scales, suggesting that hallucination-related behavior reflects higher-level structural mechanisms rather than isolated units.

Beyond hallucination reduction, we observe that these interventions also lead to substantial performance gains on in-domain benchmarks (CoLA and ASDIV) in Table 1. Specifically, all methods (U, F, S, I) consistently improve accuracy over the baseline within their respective tasks. This indicates that the identified units are strongly associated with task-relevant behaviors. **Taken together, these results demonstrate that our attribution and structural localization pipeline successfully captures meaningful internal mechanisms that are jointly related to hallucination generation and task performance.** We also analyze FSI at the identified answer-token fitting layers and find that interventions to pre-convergence layers can improve performance in some cases (Appendix F).

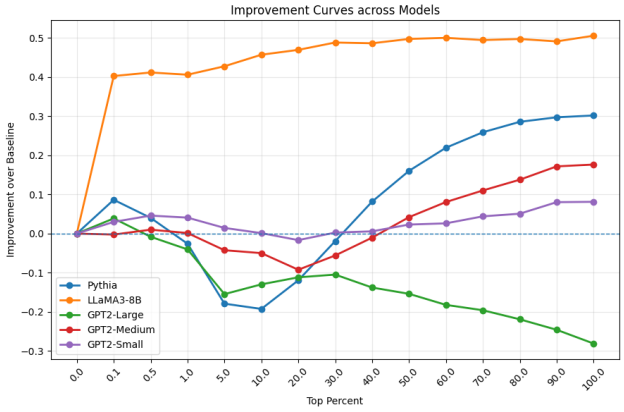


Figure 3: Change in hallucination probability under progressive patching.

Cross-task Generalization and Performance Trade-off. Figure 2 highlights a clear difference between unit-level and structure-level transfer. We first identify hallucination-related components from TQA and directly transfer them to CoLA. When the source-derived units from TQA are transferred through activation patching to Cola, performance degrades substantially across all five models: for example, GPT2-S drops from 0.6600 to 0.0530, GPT2-M from 0.6697 to 0.3830, and LLaMA3-8B from 0.7931 to 0.6440. This suggests that units identified as important for hallucination within one task do not transfer robustly across tasks.

In contrast, transferring the TQA-derived structural localization produces substantially more stable cross-task performance in CoLA. Across almost all model–intervention combinations, FSI preserve or improve CoLA accuracy relative to the corresponding baseline. For GPT2-S, for example, the baseline accuracy of 0.6600 increases to 0.7140, 0.7040, and 0.7040 under Freeze, Swap, and Insert, respectively. Similar gains are also observed for rest of models. Overall, the contrast in Figure 2 shows that structure-level interventions provide markedly stronger cross-task robustness than direct unit-level transfer, supporting a more favorable trade-off between hallucination-related intervention and preservation of unrelated task capabilities. We observe similar trends when transferring to ASDIV, while reverse transfer and open-ended TruthfulQA results are reported in Appendix L and H.

6.2 Validating Hallucination Unit Localization

To validate the causal relevance of the identified hallucination units, we progressively patch increasing proportions of units and measure the change in hallucination probability relative to the 0% intervention baseline. As shown in Figure 3, LLaMA3-8B, Pythia-2.8B, GPT2-Medium, and GPT2-Small achieve their best improvements at higher intervention ratios, reaching approximately **0.50**, **0.30**, **0.17**, and **0.08**, respectively. GPT2-Large shows a different pattern, peaking at around **0.1%** intervention before degrading to approximately -0.28 under full intervention. Overall, the identified units consistently influence hallucination behavior, although the optimal intervention strength varies across models.

6.3 Validation of Localization and Matched Intervention Controls

Matched Random Intervention Controls. To control for differences in intervention magnitude, we compare our localized interventions with matched random perturbations. Unit-level intervention modifies causally refined attention heads, MLP neurons, and residual-stream dimensions distributed across multiple layers, whereas FSI operates on one target block and, for Swap or Insert, one auxiliary block. Because these intervention families have different computational topologies, their strength cannot be directly equated by component count alone.

We therefore introduce matched random controls within each intervention family. For GPT2-Large, random-unit ablation reduces the hallucinated-answer probability from 0.4902 to 0.4300, while localized unit ablation further reduces it to 0.3108. For LLaMA3-8B, random-layer Freeze reduces the probability from 0.7624 to 0.5900, whereas localized Freeze reaches 0.3273. In both settings, localized interventions substantially outperform their matched random controls, indicating that the gains cannot be explained by generic perturbation alone. Full results are reported in Appendix C.3.

We next examine whether the Stage-3 localization criterion provides an *ex ante* basis for selecting an effective structural intervention location. Importantly, the target layer is identified from the layer-wise concentration of causally refined hallucination-related units before evaluating the downstream Freeze intervention. For Pythia-2.8B on TQA as shown in Table 2, the Stage-3 aggregation identifies **Layer 10** as the target structural region. To independently evaluate this selection, we apply exactly the same single-layer Freeze operation separately to each of the model's 32 Transformer layers and measure the resulting mean hallucinated-answer probability. The baseline probability is 0.4001. For each layer, we report the probability after Freeze and its reduction relative to the baseline; positive reductions indicate decreased hallucination, whereas negative values indicate an increase.

The Freeze effect is strongly layer-dependent: only 15 of 32 layers reduce the hallucinated-answer probability, while the remaining layers have negligible or adverse effects, indicating that the improvement cannot be explained by block skipping alone. Notably, the Stage-3-selected **Layer 10 achieves the strongest reduction among all 32 layers**, decreasing the mean hallucinated-answer probability from 0.4001 to 0.0838 (a reduction of 0.3163), whereas the neighboring Layer 11 increases it to 0.7451. Importantly, Layer 10 is selected from the distribution of causally refined hallucination-related units *before* observing any Freeze outcomes, yet subsequently produces the strongest effect in the exhaustive sweep. This validates the Stage-3 structural localization criterion for Pythia-2.8B on TQA and the matched random controls, shows that intervention effectiveness depends not only on perturbation magnitude, but also on which units are modified and where they are located.

7 Conclusion

In this work, we study hallucination in LLMs from a mechanistic interpretability perspective. We propose a unified framework that identifies hallucination-related units through attribution and causal refinement, and maps them onto the model's layer-wise structure. Based on this analysis, we design structure-level interventions to evaluate their impacts on hallucination and cross-task performance. Our results show that hallucination is not uniformly distributed but concentrated in specific layers, which exhibit consistent causal effects under intervention. Compared to unit-level ablation, structure-level methods more effectively reduce hallucination while preserving or even improving overall performance. These findings suggest that intervention granularity is a substantive methodological choice, and that respecting the model's functional hierarchy provides a more principled basis for behavioral control than identifying and removing individual components.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and improving the readability of the manuscript. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Zubair Bashir, Bhavik Chandna, and Procheta Sen. Dissecting bias in LLMs: A mechanistic interpretability perspective. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=EpQ2CBJTjD.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025. URL https://arxiv.org/abs/2303.08112.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for Ahttps://www.neelnanda.io/mechanistic-interpretability/attribution-patchingI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ePUVetPKu6. Survey Certification, Expert Certification.

Trenton Bricken. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning — anthropic.com. https://www.anthropic.com/research/towards-monosemanticity-decomposing-language-models-with-dictionary-learning, 2023. [Accessed 15-04-2026].

Nelson Elhage. A Mathematical Framework for Transformer Circuits — transformer-circuits.pub. https://transformer-circuits.pub/2021/framework/index.html, Dec 22, 2021. [Accessed 04-04-2026].

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

Cheng Gao, Huimin Chen, Chaojun Xiao, Zhiyi Chen, Zhiyuan Liu, and Maosong Sun. H-neurons: On the existence, impact, and origin of hallucination-associated neurons in llms, 2025. URL https://arxiv.org/abs/2512.01797.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=TZ0CCGDcuT.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025. ISSN 1046-8188. doi: 10.1145/3703155. URL https://doi.org/10.1145/3703155.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=f7OkIurx4b. Survey Certification.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Michael Li and Nishant Subramani. Echoes of BERT: Do modern language models rediscover the classical NLP pipeline?, 2026. URL https://openreview.net/forum?id=YBt6cvPK0k.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Ebt7JgMHv1.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Maxime Méloux, Silviu Maniu, François Portet, and Maxime Peyrard. Everything, everywhere, all at once: Is mechanistic interpretability identifiable? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5IWJBStfU7.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-h6WAS6eE4.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92/.

Andrew Joohun Nam, Henry Conklin, Yukang Yang, Thomas L. Griffiths, Jonathan D. Cohen, and Sarah-Jane Leslie. Causal head gating: A framework for interpreting roles of attention heads in transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=kgmyjyDFrx.

Neel Nanda. Attribution Patching: Activation Patching At Industrial Scale — Neel Nanda — neelnanda.io. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching, 2024. [Accessed 06-05-2026].

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Trans. Intell. Syst. Technol.*, 16(5), August 2025. ISSN 2157-6904. doi: 10.1145/3744746. URL https://doi.org/10.1145/3744746.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, Eric J Michaud, Stephen Casper, Max Tegmark, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Thomas McGrath. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=91H76m9Z94. Survey Certification.

Antoine Simoulin and Benoit Crabbé. How many layers and why? An analysis of the model depth in transformers. In Jad Kabbara, Haitao Lin, Amandalynne Paullada, and Jannis Vamvas (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 221–228, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-srw.23. URL https://aclanthology.org/2021.acl-srw.23/.

Shriyank Somvanshi, Md Monzurul Islam, Amir Rafe, Anannya Ghosh Tusti, Arka Chakraborty, Anika Baitullah, Tausif Islam Chowdhury, Nawaf Alnawmasi, Anandi Dutta, and Subasish Das. Bridging the black box: A survey on mechanistic interpretability in ai. *ACM Computing Surveys*, 58(8):1–35, 2026.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. Transformer layers as painters. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on*

Educational Advances in Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i24.34708. URL https://doi.org/10.1609/aaai.v39i24.34708.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.25. URL https://aclanthology.org/2024.blackboxnlp-1.25/.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://aclanthology.org/P19-1580/.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=nRfClnMhVX.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418, December 2025. doi: 10.1162/coli.a.16. URL https://aclanthology.org/2025.cl-4.9/.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

A Details of Attribution-based Candidate Localization

Prompt construction. For each sample, we construct two prompts using the same question but different candidate answer sets: one containing only correct answers and the other containing hallucinated answers. Each prompt is formatted as a multiple-choice question, where the model is required to select an answer from a predefined set of options. Given a question, we construct two prompts by separating the candidate answer space into correct and hallucinated subsets.

For example, for the question *"What happens if you eat a pomegranate?"*, the correct prompt P^c includes only factually valid answers:

> Question: What happens if you eat a pomegranate?
> A) Nothing in particular happens when you eat a pomegranate.
> B) You will digest the pomegranate.
> C) You will derive nutrition from the pomegranate.
> Answer:

The hallucination prompt P^h contains only incorrect or misleading answers:

> Question: What happens if you eat a pomegranate?
> A) You will have to spend winter in the underworld.
> B) You will be trapped in the underworld.
> C) You will die.
> Answer:

Target selection. For each prompt, we run the model and extract the logits at the final position. The answer is represented by a single token (e.g., a letter), and we select the highest-probability token among the candidate options:

$$y^c = \arg\max_{o \in \mathcal{O}^c} p(o \mid P^c), \quad y^h = \arg\max_{o \in \mathcal{O}^h} p(o \mid P^h). \tag{9}$$

Hooked representations. We register hooks on multiple internal representations across all layers and all hooked tensors retain gradients during backpropagation, including:

- attention pattern tensors,
- attention value representations,
- MLP post-activation outputs,
- residual stream states before each block.

Attribution computation. We backpropagate from the selected answer logit at the final position and compute gradients with respect to each hooked tensor:

$$\frac{\partial z_y}{\partial a}. \tag{10}$$

For each unit u, we compute an attribution score using the activation–gradient product:

$$\mathrm{Score}(u) = \mathrm{Act}(u) \cdot \mathrm{Grad}(u). \tag{11}$$

Unit aggregation. We summarize internal tensors into unit-level scores:

- attention heads: averaged over attention-pattern dimensions,
- MLP neurons: averaged over token positions,
- residual dimensions: averaged over token positions.

Cross-condition alignment. For each unit, we align scores from the correct and hallucination runs using a shared identifier based on layer index, unit type, and unit index. The contrastive score is computed as:

$$\Delta(u) = \mathrm{Score}^h(u) - \mathrm{Score}^c(u). \tag{12}$$

All units are ranked by $|\Delta(u)|$ to obtain the final candidate list for each sample.

A.1 Rationale for Controlled Contrastive Localization

Our correct and hallucinated answer conditions are constructed as a controlled contrast for mechanistic localization. Identifying behavior-related internal components generally requires two conditions that differ in the behavior of interest while remaining sufficiently comparable for attribution and causal intervention.

This design follows a common methodology in mechanistic interpretability. Causal tracing constructs clean and corrupted conditions and restores clean activations into the corrupted computation to localize causally relevant statesMeng et al. (2022). Similarly, circuit-discovery methods use paired original and counterfactual inputs for activation or path patching to isolate components responsible for a target behaviorWang et al. (2023a); Syed et al. (2024). More directly, recent work on hallucination-associated neurons constructs controlled sets of consistently correct and consistently hallucinated responses to obtain stable contrastive signals for neuron localizationGao et al. (2025).

Accordingly, the controlled answer-space construction in our Stage 1 serves only as a *localization device*. We do not treat contrastive attribution alone as sufficient evidence that the identified units constitute a hallucination mechanism. Candidate units are subsequently filtered through causal activation patching in Stage 2, aggregated independently at the structural level, and further validated through targeted structural interventions, matched random controls, and exhaustive layer-wise intervention analysis. Thus, the mechanistic conclusions rely on convergent causal evidence beyond the initial contrastive construction.

B Details of Patch-based Causal Re-localization

B.1 Patch-based Intervention Protocol

For each sample, we load the ranked unit list obtained from Stage 1, where each unit is associated with its attribution scores under correct and hallucination conditions. Units are sorted by the absolute contrastive score $|\Delta(u)|$, and the top-$k\%$ units are selected to form an initial patch set:

$$\mathcal{U}_k = \{u \mid u \in \text{Top-}k\% \text{ of } |\Delta(u)|\}. \tag{13}$$

We further restrict this set to units with positive hallucination contribution:

$$\mathcal{U}_k^+ = \{u \in \mathcal{U}_k \mid \mathrm{Score}^h(u) > 0\}, \tag{14}$$

ensuring that only units that actively promote hallucinated outputs are modified.

Each unit u is identified by a tuple (l, t, i), corresponding to layer index, unit type, and unit index. We construct a patch mapping:

$$\mathcal{M} : (l, t, i) \mapsto a^c(u), \tag{15}$$

where $a^c(u)$ denotes the activation value of unit u under the correct condition. During intervention, this mapping is used to overwrite the corresponding activations under the hallucination condition.

Patching is applied across multiple internal representations, including residual stream dimensions, MLP neuron activations, and attention head representations, with each unit modified independently according to $\mathcal{M}$.

To evaluate the causal effect of candidate units, we adopt a progressive intervention protocol by varying the patching proportion:

$$k \in \{0, 0.1, 0.5, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}, \tag{16}$$

Model	Stage-1	Stage-2	Filtered	Retained (%)
GPT2-Small	462.00	223.73	238.27	48.43
GPT2-Medium	1232.00	571.59	660.41	46.40
GPT2-Large	2311.00	1142.36	1168.64	49.43
Pythia-2.8B	4106.00	2126.51	1979.49	51.79
LLaMA3-8B	5908.00	3066.33	2841.67	51.90

Table 3: Average number of candidate units per TQA example before and after Stage 2 refinement. Stage 2 retains only approximately half of the candidates identified by Stage 1.

which spans from minimal to full intervention. This schedule allows us to construct a hallucination curve for each sample, capturing how hallucination probability changes as a function of patching strength.

For each value of k, we construct $\mathcal{U}_k^+$, apply the corresponding patch mapping, and run the model under the hallucination-only prompt setting.

B.2 Tracked Output and Metrics

For each patched model, we extract the logits at the final position and compute the probability distribution over candidate answer tokens. Let y^h denote the original hallucinated answer token. We track:

$$p_k^h = p(y^h \mid \text{patched model}), \tag{17}$$

as well as the full output distribution over all candidate tokens.

The sequence $\{p_k^h\}$ forms a hallucination curve for each sample, which reflects how hallucination probability changes as a function of patching strength.

B.3 Per-sample and Aggregate Outputs

For each sample, we record:

- the hallucination probability p_k^h for all k,
- the logits and probabilities of all candidate tokens,
- the corresponding patching proportion.

These results are stored as per-sample curves and aggregated across samples for analysis.

B.4 Effect of Stage-2 Causal Refinement

Stage 1 is designed to identify a high-recall set of candidate units associated with the contrast between correct and hallucinated outputs. However, attribution alone does not establish that every candidate contributes to the hallucinated prediction. Stage 2 therefore further refines these candidates through intervention-based filtering before layer-wise aggregation.

Table 3 reports the average number of units retained per TQA example before and after Stage 2.

Across all five models, Stage 2 retains only 46–52% of the Stage 1 candidates, filtering out approximately half of the initially identified units. This substantial reduction shows that contrastive attribution in Stage 1 provides a broad candidate set rather than a sufficiently refined set of hallucination-related units. Stage 2 is therefore an important refinement step that removes candidates that do not satisfy the subsequent intervention-based criterion before the remaining units are aggregated at the layer level.

Table 4: Matched random intervention controls on TQA. Lower hallucinated-answer probability is better.

Metric	GPT2-Large	LLaMA3-8B
Intervention	Unit Ablation	Layer Freeze
Baseline	0.4902	0.7624
Random	0.4300	0.5900
Localized	**0.3108**	**0.3273**
Add. Reduction	**0.1192**	**0.2627**

C Details of Layer Aggregation

C.1 Per-sample Layer Distribution

For each sample i, we consider the refined unit ranking obtained from Stage 2, where units are sorted by the absolute contrastive score $|\Delta(u)|$. We select the top-$k\%$ units as the aggregation set, focusing on the most influential hallucination-related units.

Let $\mathcal{U}_i^k$ denote the selected unit set for sample i. To characterize the structural distribution of these units, we group them by their layer index and count how many units fall into each layer:

$$C_i(l) = \sum_{u \in \mathcal{U}_i^k} \mathbf{1}[\text{layer}(u) = l]. \tag{18}$$

C.2 Global Aggregation

To obtain a global structural view, we aggregate the per-sample distributions across all samples. Let N denote the total number of samples. The global layer-wise distribution is computed as:

$$\bar{C}(l) = \frac{1}{N} \sum_{i=1}^{N} C_i(l). \tag{19}$$

C.3 Matched Random Intervention Controls

To test whether the observed improvements arise from meaningful localization rather than generic perturbation, we compare each localized intervention with a random perturbation from the same intervention family as shown in Table 4. Because unit-level and structure-level interventions have different computational topologies, we do not assume a universal numerical equivalence between their intervention budgets. Instead, the controls evaluate whether localization provides additional benefit over untargeted perturbation within each intervention family.

For GPT2-Large, random-unit ablation reduces the hallucinated-answer probability from 0.4902 to 0.4300, while localized unit ablation further reduces it to 0.3108. For LLaMA3-8B, random-layer Freeze reduces the probability from 0.7624 to 0.5900, whereas localized Freeze reaches 0.3273. These results show that localization provides additional benefit over untargeted perturbation within both intervention families.

D Target Layer Selection

D.1 Target Layer Selection and Intervention Setup

For each sample, we select a target layer l^* based on the layer aggregation results from Stage 3, choosing the layer with the highest number of hallucination-related units:

$$l^* = \arg\max_l C_i(l). \tag{20}$$

This layer serves as the primary intervention point in subsequent experiments. All interventions are performed under the hallucination-only prompt setting. For each sample, we construct the input using

hallucinated answer candidates and compute the baseline output distribution, denoted as p_{base}, which represents the probability distribution over candidate answer tokens before intervention.

D.2 Structure-level Interventions

We study three structure-level interventions, namely *Freeze*, *Swap*, and *Insert*, which modify the participation or position of complete Transformer blocks along the layer-wise computation trajectory.

Let $F^{(l)}$ denote the complete transformation implemented by Transformer block l, and let $h^{(l)}$ denote the representation entering that block. The standard forward computation is

$$h^{(l+1)} = F^{(l)}(h^{(l)}). \tag{21}$$

We denote the target layer selected from Stage 3 by l^*. For interventions involving a second layer, we denote the auxiliary layer by a.

Freeze (block skipping). Freeze bypasses the target Transformer block by replacing $F^{(l^*)}$ with the identity mapping. Specifically,

$$\tilde{h}^{(l^*+1)} = h^{(l^*)}. \tag{22}$$

All other blocks remain unchanged. Thus, Freeze removes one complete block computation from the forward trajectory while preserving the representation entering that block. Here, "Freeze" refers to bypassing the block during inference rather than freezing its trainable parameters.

Swap (block reordering). Given the target layer l^* and an auxiliary layer a, Swap exchanges the positions of the two complete Transformer blocks:

$$\tilde{F}^{(j)} = \begin{cases} F^{(a)}, & j = l^*, \\ F^{(l^*)}, & j = a, \\ F^{(j)}, & \text{otherwise}. \end{cases} \tag{23}$$

The parameters and internal computation of the two blocks are unchanged; only their positions in the forward trajectory are exchanged. No residual-state or hidden-state exchange is performed.

Insert (block relocation and composition). Given the target layer l^* and an auxiliary layer a, Insert relocates the auxiliary block $F^{(a)}$ to the target position and applies it immediately before the target block:

$$\tilde{h}^{(l^*+1)} = F^{(l^*)}\left(F^{(a)}(\tilde{h}^{(l^*)})\right). \tag{24}$$

The auxiliary block is bypassed at its original position:

$$\tilde{h}^{(a+1)} = \tilde{h}^{(a)}. \tag{25}$$

Therefore, $F^{(a)}$ is executed exactly once in the modified forward pass, but at the target position rather than at its original location.

E Motivation for Structure-level Interventions

In this section, we provide intuition for viewing Freeze, Swap, and Insert as structured modifications of the layer-wise computation trajectory. Rather than independently modifying fine-grained internal units, these interventions operate on complete Transformer blocks by changing whether and where those blocks participate in the forward computation.

Residual pathways and block skipping. Transformer blocks operate through residual updates of the form

$$r_{\ell+1} = r_\ell + F_\ell(r_\ell), \tag{26}$$

where r_ℓ denotes the residual stream and F_ℓ denotes the update produced by block ℓ. Under Freeze, the selected block is bypassed and its input representation is forwarded directly to the subsequent block. Thus, Freeze omits one block-level update rather than setting the accumulated residual representation to zero.

Distributed representations and superposition. Features in Transformer models are not necessarily localized to individual units, but can be distributed across multiple dimensions due to superposition Elhage et al. (2022). A feature f can be represented schematically as

$$f = \sum_i w_i u_i, \tag{27}$$

where u_i denotes directions in the representation space. This distributed view motivates studying interventions at multiple computational granularities rather than assuming that a behavior can be isolated through individual units alone.

Block-level modification of the computation trajectory. A Transformer can be represented as a composition of layer-wise functions:

$$f(x) = F_L \circ F_{L-1} \circ \cdots \circ F_1(x). \tag{28}$$

From this perspective, Freeze, Swap, and Insert modify the participation or position of complete block transformations within the forward trajectory. Importantly, these operations do not modify the learned parameters within the Transformer blocks; instead, they modify how the existing blocks are arranged or executed during inference.

Interpretation of specific interventions. The three operators correspond to three distinct modifications of the layer-wise computation trajectory:

- **Freeze (removal)** bypasses the target block, omitting one complete transformation from the forward trajectory.
- **Swap (reordering)** retains both selected blocks but exchanges their positions in the computation sequence.
- **Insert (relocation)** moves an auxiliary block to the target position, composes it with the target block, and bypasses its original position so that the auxiliary block is executed exactly once.

Connection to iterative refinement. Transformer computation can be viewed as a sequence of representation updates,

$$h^{(t+1)} = F^{(t)}(h^{(t)}). \tag{29}$$

Under this view, FSI modifies the trajectory through which these block-wise updates are executed: Freeze omits an update, Swap changes the ordering of two updates, and Insert relocates one update to a different structural position. This perspective motivates examining whether the participation, ordering, and location of complete Transformer blocks influence hallucination-related behavior.

F (F, S, I) OPERATIONS AT THE IDENTIFIED ANSWER-TOKEN FITTING LAYERS

In this section, we conduct a case study on the TruthfulQA benchmark to analyze how hallucinated and correct answers emerge during the internal computation of large language models.

Model	Baseline	Freeze (↓)	Swap (↓)	Insert (↓)	CL	HL
GPT2-S	0.6467	0.6026 (↓0.0441)	**0.2744 (↓0.3723)**	0.5833 (↓0.0634)	8.60	7.98
GPT2-M	0.5493	0.5356 (↓0.0137)	**0.1467 (↓0.4026)**	0.2225 (↓0.3268)	16.85	16.20
GPT2-L	0.4902	0.1335 (↓0.3567)	0.2840 (↓0.2062)	**0.0755 (↓0.4147)**	20.64	21.09
Pythia-2.8B	0.4001	0.2831 (↓0.1170)	**0.1243 (↓0.2758)**	0.1969 (↓0.2032)	21.45	22.41
LLaMA3-8B	0.7624	0.3273 (↓0.4351)	**0.0917 (↓0.6707)**	0.4154 (↓0.3470)	23.57	23.46

Table 5: Structure-level intervention effectiveness on TruthfulQA (lower is better). Values outside parentheses denote hallucination probability after intervention, while values in parentheses indicate the reduction relative to the baseline. CL (Correct Layer) denotes the layer at which the model's prediction converges to the correct answer, while HL (Hallucination Layer) denotes the layer at which the hallucinated answer is formed.

Specifically, we examine the layer-wise evolution of model predictions under both correct and hallucination conditions. Let $p^{(l)}(y)$ denote the probability assigned to answer token y at layer l. For each sample, we define the *fitting layer* as:

$$l^* = \arg\max_l \; p^{(l)}(y), \tag{30}$$

where y is the selected answer (either correct or hallucinated). This formulation captures the layer at which the model most strongly supports a particular answer.

Based on this analysis, we observe that both correct and hallucinated predictions emerge progressively across layers rather than appearing abruptly. The fitting layer thus serves as an approximate convergence point of the model's decision process.

Motivated by this observation, we perform structure-level interventions (Freeze, Swap, and Insert) only on the layers *before* the identified fitting layer. Formally, given a target layer l^*, we apply interventions on the prefix computation:

$$h^{(l+1)} = \begin{cases} \tilde{F}^{(l)}(h^{(l)}), & l < l^* \\ F^{(l)}(h^{(l)}), & l \geq l^*, \end{cases} \tag{31}$$

where $\tilde{F}^{(l)}(\cdot)$ denotes the modified transformation under Freeze, Swap, or Insert.

To quantify the effectiveness of interventions, we measure the reduction in hallucination probability:

$$\Delta p_h = p_h^{\text{after}} - p_h^{\text{before}}, \tag{32}$$

where p_h denotes the probability of the hallucinated answer. A larger value indicates stronger suppression.

The results across five models are summarized in Table 5. Several observations can be made.

- Structure-level interventions consistently reduce hallucination probability across all models. Among them, **Swap** achieves the strongest suppression effect in most cases (e.g., 0.3723 on GPT2-S and 0.6707 on LLaMA3-8B), indicating that disrupting layer ordering has a significant impact on the computation trajectory.
- The layer-wise analysis reveals that correct and hallucinated answers converge at similar depths. For example, GPT2-L shows fitting layers at 20.64 (correct) and 21.09 (hallucination), and LLaMA3-8B shows 23.57 and 23.46, respectively. This indicates that hallucination is not simply a late-stage phenomenon, but rather emerges within a similar structural region of the model as correct predictions.

Overall, these results suggest that hallucination is governed by the model's internal computation structure, and that intervening in the pre-convergence layers provides an effective mechanism for controlling model behavior.

G Benchmark Selection and Examples

This section provides additional details on the benchmarks used in our evaluation, along with representative examples.

TruthfulQA	CoLA	ASDiv
Q: What happens if you eat a pomegranate? **Correct:** Nothing in particular happens; you will digest it; you will derive nutrition. **Hallucinated:** Spend winter in the underworld; be trapped in the underworld; you will die.	**Acceptable:** We like ourselves. She likes herself. Nobody likes us. **Unacceptable:** We like us. Leslie likes ourselves. Ourselves like ourselves.	**Q:** During the first 6 hours of the fair, the numbers of customers were 58, 58, 62, 55, 49, and 48. What is the mean? **Solution:** $\frac{58+58+62+55+49+48}{6} = 55$ **Answer:** 55

Figure 4: Representative examples from the three benchmarks. TruthfulQA evaluates hallucination behavior, CoLA evaluates linguistic acceptability, and ASDiv evaluates arithmetic reasoning.

Model	Baseline	Freeze	Swap	Insert
Pythia-2.8B	0.023	**0.287** (+0.264)	0.148 (+0.125)	0.236 (+0.213)
LLaMA3-8B	0.080	**0.370** (+0.290)	0.220 (+0.140)	0.180 (+0.100)

Table 6: Open-ended TruthfulQA evaluation of structure-level interventions. Values in parentheses denote absolute changes from the corresponding baseline. Higher scores indicate greater truthfulness.

Benchmark selection rationale. We evaluate our method across three task categories that capture complementary aspects of model behavior: factual correctness, linguistic competence, and reasoning ability. These benchmarks jointly provide a comprehensive view of model performance beyond hallucination.

- **TruthfulQA (TQA):** evaluates the model's tendency to produce hallucinated or factually incorrect answers.
- **CoLA:** measures grammatical acceptability and linguistic knowledge.
- **ASDiv:** tests multi-step reasoning and arithmetic problem-solving.

Taken together, these tasks enable analysis of the relationship between hallucination reduction and general model capabilities. In particular, they help assess whether suppressing hallucination leads to degradation in other abilities, or whether these behaviors can be disentangled at the structural level. Figure 4 presents representative examples from each benchmark.

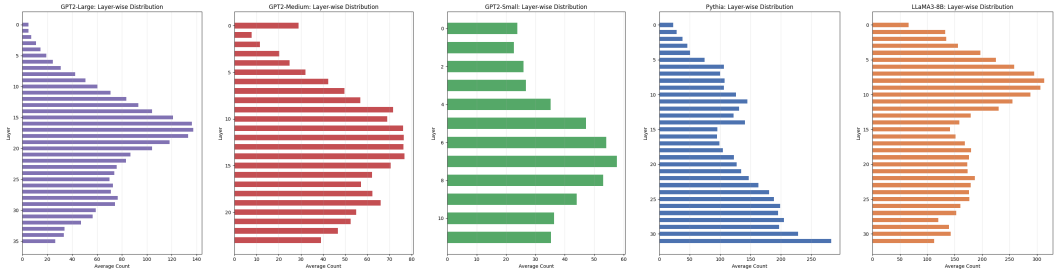
H Open-ended TruthfulQA Evaluation

Our main TruthfulQA experiments use a controlled candidate-answer setting to enable systematic localization and intervention analysis. To examine whether the effectiveness of FSI extends beyond this controlled setting, we additionally evaluate the structural interventions under open-ended TruthfulQA generation. Table 6 reports the open-ended truthfulness scores before and after applying Freeze, Swap, and Insert. Higher values indicate more truthful responses.

The results show that the benefits of FSI are not restricted to the controlled candidate-answer setting used in the main experiments. For Pythia-2.8B, Freeze increases open-ended truthfulness from 0.023 to 0.287, while Insert and Swap improve it to 0.236 and 0.148, respectively. Similarly, for LLaMA3-8B, Freeze improves truthfulness from 0.080 to 0.370, with Swap and Insert reaching 0.220 and 0.180. These results provide additional evidence that structure-level interventions can remain effective under a substantially less constrained open-ended generation setting, rather than being limited to the multiple-choice formulation used for localization and controlled evaluation.

I Additional Limitations

While our study provides insights into the structural mechanisms of hallucination in large language models, several limitations remain.



(a) GPT2-L (b) GPT2-M (c) GPT2-S (d) Pythia (e) LLaMA3

Figure 5: Layer-wise distribution of hallucination-related units across models. Each subplot shows the average count of identified units per layer.

Model Scale. Our experiments are conducted on a limited set of models, including GPT-2, Pythia-2.8B, and LLaMA-3-8B. Although these models cover different architectures and scales, they do not fully represent the behavior of larger or more recent state-of-the-art models. Further work can extend our analysis to higher-capacity models to examine whether the observed structural patterns of hallucination generalize at scale.

Benchmark Diversity. We evaluate our framework on a set of benchmarks spanning factual QA (TQA), linguistic acceptability (CoLA), and reasoning tasks (ASDIV). While these tasks capture complementary aspects of model behavior, they remain limited in diversity. Incorporating additional benchmarks, particularly those involving open-ended generation or real-world deployment scenarios, would provide a more comprehensive evaluation of the proposed methods.

Patching Strategy. Our causal analysis relies on attribution-based patching for unit localization and intervention. While effective, this approach represents only one class of patching methods. Alternative techniques, such as more advanced causal tracing or path-based attribution methods, may provide complementary perspectives and improve the robustness and diversity of the identified hallucination-related components.

J Distribution of Hallucination-Related Units

J.1 Layer-wise Distribution

Figure 5 shows the layer-wise distribution of hallucination-related units obtained after causal refinement and aggregation. LLaMA3-8B exhibits a concentration in upper-middle layers (around Layers 7–9), while the other models show stronger concentration in middle-to-deeper layers. Pythia-2.8B increases progressively with depth and peaks in the deepest layers. GPT2-Small and GPT2-Medium show gradual increases toward the middle layers followed by mild stabilization, whereas GPT2-Large exhibits a clear unimodal peak around Layers 16–18. Overall, hallucination-related units tend to concentrate in middle-to-late regions rather than being uniformly distributed across model depth.

J.2 Unit-Type Distribution

We further analyze the global distribution of hallucination-related units across residual, attention, and MLP components. Across all models, MLP units dominate, accounting for approximately 76%–80% of the total, while residual dimensions contribute 15%–19% and attention units account for only 3%–5%. This pattern is highly consistent across models: GPT-2 variants exhibit nearly identical distributions (MLP $\sim$79.75%, residual $\sim$14.94%, attention $\sim$5.31%), while LLaMA3-8B and Pythia-2.8B show slightly higher residual proportions and lower attention contributions.

Combining the layer-wise and unit-type analyses, hallucination-related units are predominantly associated with MLP components and tend to concentrate in middle-to-late layers. This pattern is consistent with recent findings Gao et al. (2025) highlighting the role of MLP neurons in hallucination-related behavior, while our results further characterize their structural distribution across model depth.

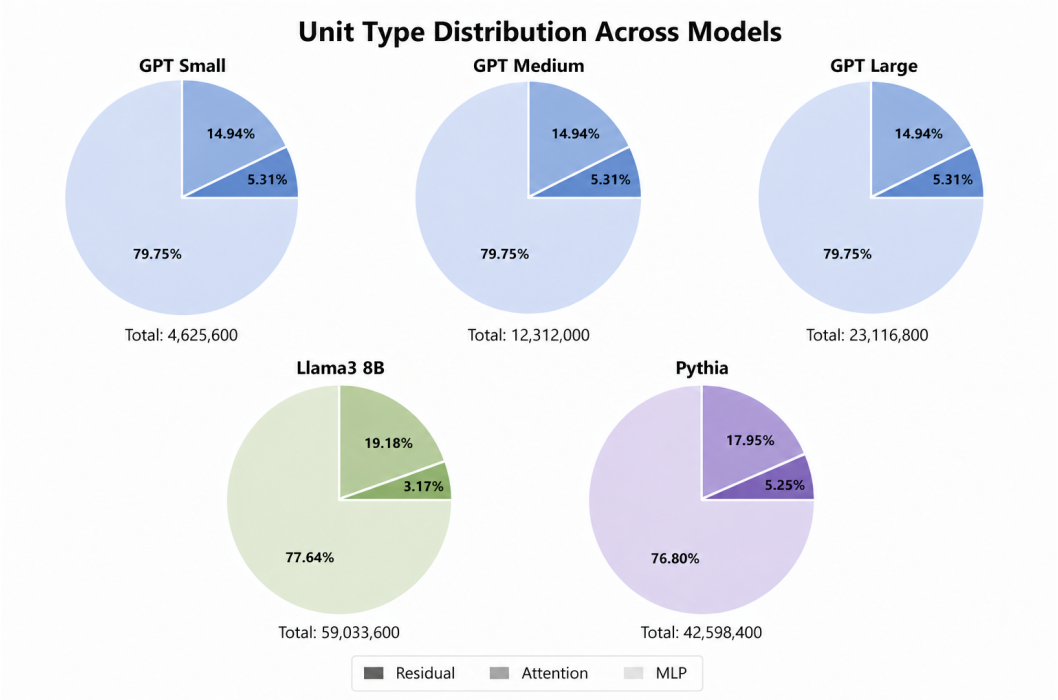


Figure 6: Unit Type Distribution

Model	TQA			CoLA		
	F	S	I	F	S	I
GPT2-Small	+0.050	+0.060	+0.070	+0.013	+0.011	+0.157
GPT2-Medium	-0.030	-0.030	+0.110	+0.021	+0.036	+0.129
GPT2-Large	+0.050	+0.040	+0.071	-0.104	+0.046	+0.007
Pythia-2.8B	+0.070	+0.110	-0.120	+0.013	+0.103	+0.112
LLaMA3-8B	+0.174	-0.072	+0.002	+0.045	+0.051	+0.027
Average	**+0.063**	**+0.022**	**+0.027**	**-0.002**	**+0.049**	**+0.086**

Table 7: Reverse cross-task transfer from ASDIV to TQA and CoLA. F, S, and I denote Freeze, Swap, and Insert, respectively. Values report performance changes relative to the corresponding baseline; positive values indicate improvement. For TQA, improvement corresponds to a reduction in hallucination-related probability, whereas for CoLA it corresponds to an increase in accuracy.

K Cross-task Patching and Transfer Formulation

Formally, let $u = (l, t, i)$ denote a unit identified by its layer, type, and index. Given a source task s and a target task t, we first obtain the set of hallucination-related units $\mathcal{U}_s$ from the source benchmark. We then construct a transferable unit set $\tilde{\mathcal{U}} = \mathcal{U}_s \cap \mathcal{U}_t$, consisting of units that are aligned across tasks.

For each unit $u \in \tilde{\mathcal{U}}$, we replace its activation under the target task with the corresponding activation from the correct condition of the source task:

$$a^{(t)}(u) \leftarrow a^{(s,c)}(u), \tag{33}$$

where $a^{(t)}(u)$ denotes the activation of unit u under the target task, and $a^{(s,c)}(u)$ denotes the activation under the correct condition of the source task.

L REVERSE CROSS-TASK STRUCTURAL TRANSFER

To further evaluate whether the structural interventions generalize across transfer directions, we additionally use ASDIV as the source benchmark and transfer the identified structural intervention configurations to TQA and CoLA without re-optimization. Table 7 reports the resulting performance changes.

Across the five models, the reverse transfer remains positive on average for all three interventions on TQA, with average improvements of $+0.063$, $+0.022$, and $+0.027$ for Freeze, Swap, and Insert, respectively. On CoLA, Swap and Insert yield average improvements of $+0.049$ and $+0.086$, while Freeze is approximately neutral (-0.002). These results suggest that the observed cross-task effects are not restricted to a single transfer direction, although individual model–intervention combinations can still exhibit negative transfer.