Conceptual Hierarchies within LLMs

Tiago Almeida, Zining Zhu, Yue Ning
Stevens Institute of Technology
{talmeida, zining.zhu, Yue.Ning}@stevens.edu

Abstract

While it is widely agreed that large language models (LLMs) store concepts of multiple semantic hierarchies, much remains unknown regarding the structure of this storage. The correspondence between the functional roles of LLM components and the semantic hierarchies of knowledge remains underexplored in the current literature. For example, is information organized hierarchically within layers of an LLM? We take an initial step towards causally examining the correspondence between hierarchical concepts and the multi-granular structures (layers and attention heads) of various LLM models. Specifically, we generate a dataset of semantic hierarchies and investigate their storage locations in six LLMs using activation patching, a causal intervention technique. At the layer level, our findings show a moderate indication that concepts at finer levels of granularity are stored around 61-78% of the time ($p < 0.01$) before those at coarser granularity. There is evidence for this trend at the attention level; however, the high variability in attention level results suggests that concepts are stored across attention heads rather than within. Our results offer insight into semantic organization within LLMs.

1 Introduction

Modern large language models (LLMs) display significant linguistic capabilities, ranging from syntactic processing and semantic comprehension to encoding factual knowledge (Dai et al., 2022; Dalvi et al., 2022; Tenney et al., 2019). Mechanistic interpretability (MI) is an area of research that aims to reverse-engineer the internal mechanisms of complex machine learning models in order to explain the behavior and capabilities of AI systems (Bereska and Gavves, 2024; Geiger et al., 2021; Meng et al., 2022). MI facilitates error correction and output steering, which is crucial for developing safe and trustworthy models (Hendrycks and Mazeika, 2022; Conmy et al., 2023). A critical MI task is abstraction: how LLMs encode, process, and reason with conceptual hierarchies, which reveals the core mechanisms behind language understanding (Regneri et al., 2024).

Existing literature has explored abstraction within LLMs. Tenney et al. (2019) and Dalvi et al. (2022) both established that lexical and syntactic concepts are processed and stored before semantic concepts. Regneri et al. (2024) discovered that when prompted by hypernym and non-hypernym datasets, BERT's resulting attention matrices are differentiable. However, their evidence is correlational rather than causal. On the other hand, causal methods (Aljaafari et al., 2024; Hong et al., 2025; Meng et al., 2022) have not investigated the organization of semantic hierarchies. In other words, there is a current lack of research that causally explores LLMs' storage of multi-granular concepts.

This paper aims to examine conceptual hierarchies within LLMs using a causal method, activation patching. We pose the question: "Are concepts at different levels of granularity stored hierarchically at the layer and attention levels within language models?" We generate a dataset of hierarchical concepts from WordNet (Miller, 1994) using Grok-3 (xAI, 2024), a model that outperformed Gemini 2.0, GPT4, and Claude 3.5 on various reasoning tasks and is suitable for creating a comprehensive dataset. We then apply activation patching (Ghandeharioun et al., 2024; Zhang and Nanda, 2024) on this dataset in order to intervene in the components of six models, revealing the causal relation between these patched components and the storage of conceptual hierarchies (Vig et al., 2020). Inspired by the metrics proposed by Tenney et al. (2019) and Zhang and Nanda (2024), we derive two *center-of-gravity* metrics and conduct two statistical tests to determine the hierarchical organization of multi-granular concepts.

We present three main findings: i) finer-grained

Findings of the Association for Computational Linguistics: ACL 2026, pages 41067–41079
July 2-7, 2026 ©2026 Association for Computational Linguistics

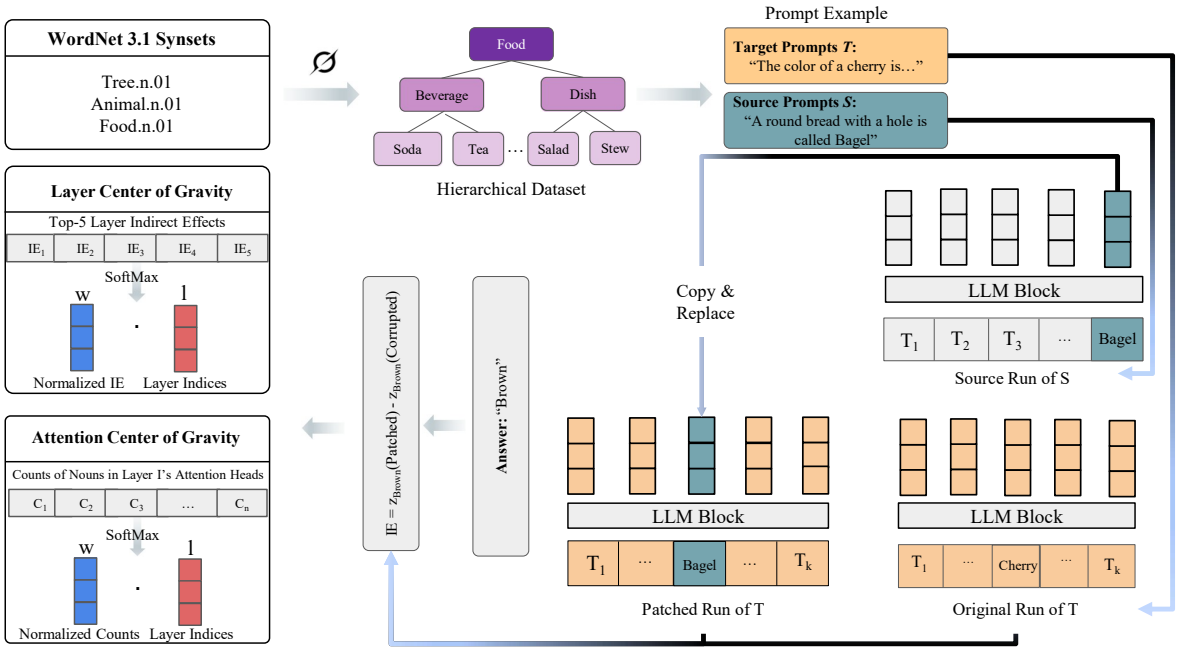

Figure 1: An overview of our approach. We first utilize NLTK and Grok-3 to generate a dataset of hierarchical concepts (top). We then patch the concepts to either the layer representations or the attention head value vectors (layer patching is pictured as an example). Each patch follows the Symmetric Token Replacement (STR) variation, featuring three forward runs: source, original, and patched. The input in the source run is the "source prompt". The input sentence in the original run is the "target prompt". In the patched run, the target prompt is passed to the model while the representation from the source run is *patched* into the corresponding layer/attention head of the model. The patched representation is thus the *mediator*, allowing us to measure the indirect effect of this intervention (Equations 1 and 2). Finally, we compute Layer and Attention level Centers of Gravity for each concept.

concepts (hyponyms) tend to emerge before coarser ones (hypernyms) in approximately 61-78% of layer-wise cases ($p < 0.01$), ii) the hierarchical organization of knowledge is both *model-* and *category*-dependent, and iii) attention-level hierarchies are present, though the variability in attention level results implies that knowledge is distributed across attention heads rather than localized. Together, these results advance our understanding of abstraction within LLMs and contribute to the development of more interpretable AI systems.

2 Related Work

Activation Patching Our causal approach builds on activation patching, a key MI technique for locating model components. We adopt the symmetric token replacement (STR) method recommended by Zhang and Nanda (2024), caching noun representations from a "source" contextual prompt, conducting "original" runs with a semantically related noun, and finally inserting the cached representation in "patched" runs to compute patching effects. Following this, we emulate Vig et al.'s (2020) *causal mediation analysis* framework by patching hidden representations while holding the model's input constant. We treat the patched representations as the "mediator" between the model's input and output, capturing causal mediation effects as defined in Vig et al. (2020). Prior works compute patching effects via probability differences between the original noun in the patched run and the related noun in a corrupted run. However, we decide to derive a logit difference, as suggested by (Zhang and Nanda, 2024), using only the target token in the original and patched run. Because of its robustness and fine-grained control (Zhang and Nanda, 2024), this logit difference is best suited to measure the *indirect effect* of the intervention, "which is crucial for studying the role of internal model components" (Vig et al., 2020). Thus, our causal patching method addresses the current gap in literature by locating semantic hierarchies in LLMs.

Patching in MI Literature MI literature displays substantial use of activation patching in order to find the location of concepts or facts. Meng

et al. (2022) used the technique to locate and edit factual associations in the mid-layers of GPT models. Wang et al. (2023) located a circuit for Indirect Object Identification, paving the way for the localization of linguistic capabilities in MI, including the circuit discoveries of Conmy et al. (2023); Lieberum et al. (2023). Extending this, Dumas et al. (2025) used patching in translation tasks to show that concepts are represented independently of language, with language processed earlier. In other MI studies, Hong et al. (2025) and Bronzini et al. (2024) examined the representation, evolution, and dynamics of *latent* concepts within LLMs. While these works do not extend to explicit semantic hierarchies, they reveal later-layer knowledge comprehensiveness, similar to Wang et al.'s (2024) findings of increasing relational complexity throughout the model. Focusing on abstraction, early work by Geiger et al. (2021) verified that a BERT-based model partially realizes the causal abstraction structure of logic in natural language using patching; however, this does not address the abstraction of *concepts*. In one of the emerging causal studies on conceptual abstraction, Aljaafari et al. (2024) established a model for examining conceptual interpretation (the mapping of lexical items to abstract concepts) and revealed that lexical information is stored before semantics. Although these works collectively take advantage of Vig et al.'s (2020) causal mediation analysis, they do not fully dissect conceptual abstraction within LLMs.

Conceptual Probing Probing, a correlational method, has provided evidence for conceptual abstraction. Probing is a technique that trains classifiers on the hidden representations of a model in order to discover whether classes of representations differ from one another (Youssef et al., 2023). Regneri et al. (2024) showed that a linear classifier can differentiate between hypernymy and non-hypernymy in attention matrices instantiated by a set of psychologically motivated noun pairs, establishing that hypernym-hyponym relations are present in BERT. We extend this work by discovering hypernym-hyponym storage order. Additionally, Dalvi et al. (2022) identified how lexical, syntactic, and semantic hierarchies of concepts changed throughout the model using clustering of BERT's contextualized representations at each hidden layer. The results demonstrated that lexical and syntactic processing occurred before the model processed semantics. Tenney et al. (2019) showed similar results by using "edge probing", training classifiers on BERT representations to extract linguistic information (part-of-speech, entities, etc.). Jawahar et al. (2019) obtained parallel findings using probing techniques, claiming that BERT's composition mimics "classical, tree-like structures". Jin et al. (2025) used the same technique to find that more complex concepts emerge at deeper layers. The limitation of Probing is that the evidence it provides is *correlational*, which inhibits the strength and validity of the resulting findings. Activation patching, on the other hand, provides stronger, causal evidence by directly intervening in the model's computation. Probing's correlational findings motivate our causal extension, which builds on the layer-wise linguistic hierarchies already studied. In short, we extend prior work on concept localization to multi-level semantic hierarchies, providing novel and stronger evidence for conceptual abstraction within LLMS.

3 Dataset

Database Selection We use WordNet 3.1's database to create the dataset. WordNet is a widely used lexical database that organizes nouns, adjectives, verbs, and adverbs into groups of synonyms (synsets). Like Park et al. (2025) and Aspillaga et al. (2021), we utilize WordNet's synsets to study conceptual hierarchies within LLMs. While Baroni and Lenci (2011)'s BLESS dataset provides a comprehensive set of hypernyms, we require hierarchies with multiple levels of granularity which are not present in BLESS.

Concept Categories The following categories are selected for our dataset: *animal* taxonomy, *tree* taxonomy, and *food*. These concepts are intrinsically hierarchical (contain a large depth and breadth of diverse hypernyms) and representative of traditional linguistic hierarchies (Rosch et al., 1976). Further, the selected categories are based on the superordinates in Rosch et al. (1976). While WordNet is English-centric and may be biased towards Western hierarchies, we mitigate these concerns by prioritizing universal concepts. As a result, the hierarchies allow for an exploration of conceptual abstraction within LLMs.

Hierarchy Construction We adhere to the following procedure to generate our dataset. Using NLTK 3.9.1, all nouns in the synsets corresponding to the three categories are extracted from the

WordNet database and selected via a simple random sample. Other synsets that are hyponyms of the category are used in order to obtain a sample with enough levels of granularity, but this selection is kept to a minimum to mitigate bias. The sampled nouns are grouped into concepts (i.e. nodes) and arranged hierarchically by Grok-3. We tested various LLMs (GPT-4, DeepSeek) and Grok-3 was the model that created the largest hierarchy with the provided nouns. Further, we find that the hierarchies curated by Grok-3 aligned with traditional semantic hierarchies (Rosch et al., 1976) rather than the hypernym-hyponym sets of WordNet 3.1, with the former being more ideal for our experiments. For example, consider the "food" category: WordNet contained "food" followed by its direct hyponyms "yolk" and "repast", but a hierarchy like "food" $\rightarrow$ "dish" is more adequate as it is more likely to be distinguished as such by the model. In other words, the Grok-3 hierarchies provide stronger evidence for our experiments as the concepts are more causally related to each other. The concepts within the hierarchies are nouns from the WordNet database but not necessarily part of the sample. Each category contains one hierarchy tree with a root as the category noun and a depth of 4. Thus, our data provide several semantic hierarchies.

Source-Target Prompt Dataset For each sampled noun within a concept, a *source* and a *target* prompts are created. The *source* prompt includes some context about the sampled noun and ends with the noun itself. Consequently, the hidden state of the sampled noun at layer ℓ contains the meaning and context of the concept and thus can be used to measure its location. In addition, the *target* prompt is meant to reveal a particular property of a different yet semantically related target noun. This "related" noun is simply one that shares a distinct and characteristic property with the source noun. The target prompts were structured in the same manner as the knowledge expressing prompts in Dai et al. (2022). Furthermore, the dataset stores the sampled noun in the source prompt, the corresponding noun in the target prompt, and the "patched" answer that occurs when the sampled noun replaces its corresponding noun in the target prompt. For example, consider the noun "turmeric". The source prompt could be "A yellow spice from a root is called turmeric", and the target prompt could be "The color of a cherry is...". Thus, the nouns "turmeric" and "cherry" would be stored, in addition to "yellow" (for details regarding the prompts used to generate the dataset, see Appendix D). As a result, the hidden representation of the source prompt noun could be automatically cached and used to replace that of the semantically related noun in the target prompt. Further, the patched answer could be used to measure the association between the concept and structure of the model.

Category	Nouns	SP	Concepts	sub-H	Avg. Breadth
Animal	500	363	140	16	8.1875
Food	300	266	55	8	6.1250
Tree	500	262	46	6	7.5000
Total	1300	892	241	30	7.2710

Table 1: Dataset Statistics. (1) Source prompts (SP) and sampled nouns (Nouns) differ as Grok-3 was unable to use all the nouns in the dataset and (2) breadth is defined as the number of direct hyponyms of the root of each sub-hierarchy (sub-H).

Validating Dataset Quality With 892 source prompts, 241 concepts, and 30 sub-hierarchies, our dataset is comprehensive enough to ensure the validity of our findings. Further, the large breadth and depth allow for robust statistical power when conducting tests on our results. Our dataset is self-validated as the code would only run if the stored nouns in the latter half of the dataset match the nouns in the actual prompts. Despite this, the dataset was completely checked by one qualified human annotator to ensure that (1) the context and noun are consistent within each source prompt, and (2) the patched answer tokens are consistent with the patched target prompt. Semantic relatedness between the source and target prompt nouns is ensured by manually creating a target prompt template. Less than 4% of answer tokens require correction. Thus, our dataset is precise and of high quality, which are essential for our experiments.

4 Methodology

4.1 Patching

We use patching at both the layer and attention level (following Wang et al. (2023)), but the metrics underlying the results at both levels are the same. We formalize the patching process as follows: Let S be the source prompt containing noun n_s, T be the target prompt containing the different yet related noun n_t, and a be the patched answer token.

Then, $x_P = \texttt{tokenize}(P)$, where $x_P \in \mathbb{R}^L$ is the input sequence of prompt P and $P \in \{S, T\}$.

Let the per-layer hidden states and per-head value vectors be $\mathbf{h}^{\ell} \in \mathbb{R}^{L \times d}$ and $\mathbf{v}^{\ell,h} \in \mathbb{R}^{L \times d_v}$, where L is the sequence length of P, d is the dimension of the hidden representations, and d_v is the dimension of the value vectors. Thus, for the source token index k of x_S, $\mathbf{h}^{\ell}_{S,k}$ is the hidden representation and $\mathbf{v}^{h}_{S,k}$ is the value vector of n_s at layer ℓ and attention head h respectively. In the source run, we pass the model the prompt S and store both $\mathbf{h}^{\ell}_{S,k}$ and $\mathbf{v}^{h}_{S,k}$. If there are multiple token indices $i_1, i_2, ..., i_c$ of the tokenized n_s, we average the hidden states and value vectors from i_1 to i_k. We also write the averaged vectors as $\mathbf{h}^{\ell}_{S,k}$ and $\mathbf{v}^{h}_{S,k}$ for simplicity. In addition, we use $\mathbf{v}^{h}$ to denote $\mathbf{v}^{\ell,h}$, the value vector of head h at layer ℓ.

Similarly, for the target token index m of x_T, $\mathbf{h}^{\ell}_{T,m}$ is the hidden representation and $\mathbf{v}^{h}_{T,m}$ is the value vector of n_t at layer ℓ and attention head h respectively. In the original run, we pass the model the prompt T and record the logit $z_a(T)$. In the patched run, we replace $\mathbf{h}^{\ell}_{T,m}$ and $\mathbf{v}^{h}_{T,m}$ with $\mathbf{h}^{\ell}_{S,k}$ and $\mathbf{v}^{h}_{S,k}$ respectively, and record the logits: $z_a(T \,|\, do(\mathbf{h}^{\ell}_{T,m} \leftarrow \mathbf{h}^{\ell}_{S,k}))$ and $z_a(T \,|\, do(\mathbf{v}^{h}_{T,m} \leftarrow \mathbf{v}^{h}_{S,k}))$ (using the do calculus notation of Pearl (2009)). If there are multiple token indices $i_1, i_2, ..., i_c$ of the tokenized n_t, then all n_t tokens are replaced.

If a concept is located in a specific layer or attention head, this patching will increase the likelihood of the answer token. Following existing research, we quantify this change with *indirect effect* where the patched representation is the "mediator" and the model's input is held constant (Vig et al., 2020). We use a logit difference (Zhang and Nanda, 2024) of the patched answer token in the "patched" run compared to the "original" run. Putting this together, the *indirect effect* at layer ℓ is:

$$\Delta^{(\ell)} = z_a(T \,|\, do(\mathbf{h}^{\ell}_{T,m} \leftarrow \mathbf{h}^{\ell}_{S,k})) - z_a(T), \quad (1)$$

and the indirect effect at attention head h is:

$$\Delta^{(h)} = z_a(T \,|\, do(\mathbf{v}^{h}_{T,m} \leftarrow \mathbf{v}^{h}_{S,k})) - z_a(T). \quad (2)$$

If a higher mediation effect is observed, we consider the patched representation $\mathbf{h}^{\ell}_{S,k}$ or $\mathbf{v}^{h}_{S,k}$ to reflect a better localization of the concept. When the patched answer noun consists of multiple tokens, we average the logit differences across these a tokens: $\frac{1}{n}\sum_{i=1}^{a} \Delta_i^{(\ell \text{ or } h)}$.

4.2 Layer Center-of-Gravity

To determine which layer each concept is stored in, we create a metric similar to Tenney et al.'s (2019) center-of-gravity metric. First, we select the top 5 layers with the highest logit difference for each source prompt. We confirm the validity of k = 5 for top-k layers by recomputing the CoGs for k = 3, 7, 10 and assessing the correlation between the new CoGs and top-5 CoG (see Appendix B). Then, we normalize the logit differences from these layers with softmax to obtain weights:

$$w_i = \frac{e^{k_i}}{\sum_{j=1}^{5} e^{k_j}}, \quad (3)$$

where $k_i = \Delta^{(\ell_i)}$ per Equation 1. Thus, we get

$$\textit{Layer center-of-gravity} = \sum_{i=1}^{5} w_i \cdot \ell_i, \quad (4)$$

where ℓ_i is the layer index of the *i*-th layer. Since source prompts and their nouns are grouped by concepts, we take a simple mean of the Centers of Gravity of the nouns N that fell under the same concept c: $\text{CoG}_c = \frac{1}{N}\sum_{n=1, n \in c}^{N} \text{CoG}_n$. Intuitively, a larger metric means that the concept is stored in higher layers.

4.3 Attention Center-of-Gravity

We also apply patching in each singular attention head of every layer. During the attention head computation, the value vector of the noun token(s) is extracted and patched. The value vectors carry content and are most suitable for patching (Vaswani et al., 2017; Geva et al., 2021). To determine which attention head stored a noun within a concept, we adhere to the following algorithm: (1) calculate the standard deviation of the attention head logit differences ($\Delta^{(h)}$ defined by Equation 2) across models and categories, (2) per prompt (i.e. per noun within a concept) within each layer, find which attention head metric $\Delta^{(h)}$ deviated from the prompt's median attention head metric by 3σ. We validated this choice by recomputing the CoGs with 2, 2.5, 3.5, 4σ while following the same protocol as the layer CoG (see Appendix C for details).

We compute attention center-of-gravity to determine in what layer's attention heads each concept is stored. First, we obtain the number of times a concept's nouns are stored in each layer's attention heads. In other words, for a concept c with N nouns, the attention head per-layer concept

count k_ℓ is defined as $k_\ell = \sum_{p=1}^{N} \sum_{h=1}^{H} \mathbb{I}(\Delta_p^{(h)} > \text{median}_p + 3\sigma)$, where I is the indicator function, p is the prompt S containing noun n_s, and H is the number of attention heads per layer. Then, counts k_1 to k_L are normalized using softmax to create weights:

$$w_i = \frac{e^{k_i}}{\sum_{j=1}^{L} e^{k_j}}, \quad (5)$$

where $k_i = k_{\ell_i}$ and L is the number of layers. Only non-zero counts are used. Thus:

$$\textit{Attention center-of-gravity} = \sum_{i=1}^{L} w_i \cdot \ell_i, \quad (6)$$

where ℓ_i is the layer index of the i-th layer. We average the noun Centers of Gravity according to $\text{CoG}_c = \frac{1}{N} \sum_{n=1, n \in c}^{N} \text{CoG}_n$. This metric differs from that of Equation 4 because the weights are concept counts for attention heads rather than logit differences for an entire layer. Intuitively, this new metric indicates in what layers' *attention heads* the concept is stored.

4.4 Implementation Details

We use the Hugging Face Transformers 4.52.4 library, PyTorch 2.7.0, and PyTorch forward hooks for our experiments, which are placed in layers for hidden states and value projection modules for value vectors. The experiment was conducted on an Nvidia A6000 GPU, and the running time across datasets for the 7-9B parameter models required an average of 12 hours whereas the 1.5-3B parameter models required an average of 1.5 hours.

4.5 Models

We use the following models in our experiment: Llama-2-7B, Llama-3.2-3B, Gemma-2-9B, Gemma-2-2B, Qwen-2.5-1.5B and Mistral-7B-v0.1. The models are selected to be moderately small in size due to computation restraints.

Llama-2-7B (Touvron et al., 2023), released in July 2023, is a decoder-only LLM trained on 2 trillion tokens intended for research and commercial use. At the time, it outperformed open-source models like MPT and Falcon on academic benchmarks (coding, mathematics, reasoning, etc.), but lagged behind closed-source models like GPT-4 and PaLM-2-L.

Mistral-7B-v0.1 (Jiang et al., 2023) is a 7.3B parameter model designed to balance performance and efficiency. It was released under an Apache 2.0 license and has no restrictions on usage. As a decoder-only LLM, it features Grouped-Query and Sliding Window attention mechanisms. Mistral-7B-v0.1 outperforms Llama-2-13B across many academic benchmarks, making it a competitive model.

The Gemma-2 (Team et al., 2024) family is a lightweight, state-of-the-art group of models intended for research and commercial use. Their architecture is based on a decoder-only transformer, with features such as Global, Local Sliding Window, and Grouped-Query attention. Gemma-2-9B outperforms Mistral-7B-v0.1 and Llama-2-7B on most NLP benchmarks and achieves the best performance for similarly sized models.

Llama-3.2-3B and Qwen-2.5-1.5B (Grattafiori et al., 2024; Yang et al., 2024) are small, decoder-only models with performance comparable to other similarly sized state-of-the-art LLMs. They are intended for research and commercial use.

Model	Category	Layer		Attention	
		Wilcoxon	Binomial	Wilcoxon	Binomial
Llama-2-7b	Animal	0.004*	0.003*	0.000*	0.000*
	Food	0.014*	0.002*	0.000*	0.002*
	Tree	0.999	0.999	0.381	0.617
Mistral v0.1	Animal	0.004*	0.278	0.000*	0.000*
	Food	0.003*	0.010*	0.063	0.333
	Tree	0.042*	0.185	1.000	1.000
Gemma-2-9B	Animal	0.000*	0.000*	0.181	0.136
	Food	0.001*	0.001*	0.988	0.957
	Tree	0.000*	0.000*	1.000	1.000
Llama-3.2-3b	Animal	0.004*	0.040*	0.000*	0.000*
	Food	0.000*	0.000*	0.000*	0.000*
	Tree	0.000*	0.000*	0.993	0.967
Gemma-2-2B	Animal	0.000*	0.000*	0.000*	0.000*
	Food	0.000*	0.000*	0.999	0.999
	Tree	0.000*	0.000*	0.292	0.814
Qwen-2.5-1.5B	Animal	0.102	0.499	0.999	0.999
	Food	0.006*	0.004*	0.999	0.999
	Tree	0.000*	0.008*	0.770	0.884

Table 2: Statistical test results for Layer-level and Attention-level analyses (Wilcoxon and Binomial p-values). We tested whether finer-grained concepts are stored before broader ones. Significant results with p-values < 0.05 are marked with $*$.

5 Results

To find evidence of concepts being stored hierarchically within the model, we conduct two statistical tests: a Wilcoxon Signed-Rank test and a Binomial test. The Wilcoxon Signed-Rank Test assesses whether the median difference between paired data is zero (Hollander et al., 2013; Wilcoxon, 1945), while the Binomial test determines whether the proportion of successes in a sample matches the expected proportion in the binomial distribution (Conover, 1999). The Wilcoxon Signed-Rank test

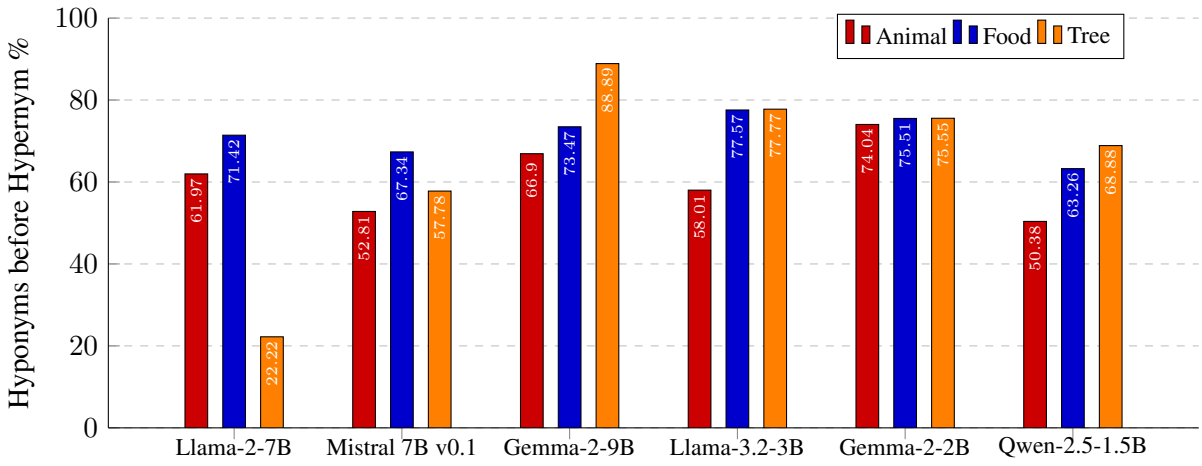

Figure 2: Layer Level: Percentage of hyponyms being stored before hypernyms across categories for six models. All plotted on the same axis for easier comparison.

is limited in that it assumes symmetry around a median; to mitigate this, we ensured that $n > 30$ for a robust test statistic (Hollander et al., 2013) and supported the analysis with the binomial test. For both tests, we consider each category sub-hierarchy, with the root as the hypernym concept and the direct sub-nodes as the hyponyms. For the Wilcoxon Signed-Rank test, the paired differences are defined as the difference between the hypernym's and hyponym's center-of-gravity. For the Binomial test, a success is defined as the hyponym's index being smaller than the hypernym's index. We also calculate the proportion of hyponyms whose center-of-gravity is smaller than that of their hypernym (see Figure 2). Table 2 displays the p-values for both the layer and attention levels. Together, these statistics provide evidence for hierarchical organization of concepts at different levels of granularity.

At the layer level, the statistical tests suggest that concepts are organized hierarchically within LLMs. As shown in Figures 2, the percentage of hyponyms whose index occurred before that of their hypernym mostly range from 61-78%. There are several outliers to this trend: the Llama-2-7B tree category (.999 p-value for both tests), Mistral 7B v0.1 animal category (.278 p-value for Binomial test), Mistral 7B v0.1 tree category (.185 p-value for Binomial test), and Qwen-2.5-1.5B animal category (.102 and .499 p-values for Wilcoxon and Binomial tests, respectively). However, the majority of cases, specifically 16/18 and 14/18 for Wilcoxon and Binomial tests, respectively, at $p < 0.05$, indicate that concepts at a finer level of granularity precede those at a coarser level. This causal evidence is consistent with Jin et al. (2025); Wang et al. (2024) as our findings also reveal that complexity, particularly that of concepts and relations, emerges at later layers. Further, the fact that multi-granular concepts are arranged hierarchically is compatible with the findings of Aljaafari et al. (2024); Jawahar et al. (2019); Tenney et al. (2019), which show that lexical concepts are stored before semantic ones.

One important observation to make is that the results vary across both models and categories. Within a model, the three categories show different percentages of finer-grained concepts occurring before more abstract concepts. For example, in Gemma-2-9B, the percentage for the animal category is 66.90%, for the food category is 73.47%, and for the tree category is 88.89%. Within a category, the percentage of hyponyms occurring before a hypernym is also different across the three models. For example, in the animal category, the percentage for Llama-2-7B is 61.97%, for Mistral 7B v0.1 is 52.81%, for Gemma-2-9B is 66.90%, for Llama-3.2-3B is 58.01%, for Gemma-2-2B is 74.04%, and for Qwen-2.5-1.5B is 50.38%. Thus, we can make the claim that the storage of conceptual hierarchies within LLMs is *category* and *model* specific.

The attention level results are different from those at the layer level (see Appendix A). For Llama-2-7B and Llama-3.2-3B, the evidence of hierarchical storage is similar to that of the layer level, but there is no indication of attention level conceptual hierarchies in the rest of the models. To associate semantic and structural hierarchies, we test whether the attention center-of-gravity (finer-grained) occurred before the layer center-of-gravity (coarser-grained) for each concept using a paired

t-test. For the model-category pairs, 8/18 resulted in $p < 0.01$ and 10/18 in $p > 0.999$ (See Appendix A for details). We hypothesize that there should be a clear trend in the paired t-test results. Either there is an association between semantic and structural hierarchies (i.e., $p < 0.01$) or there is no difference between the hierarchies at the two levels (i.e. $p > 0.05$). The latter might be the case because the layer level contains the attention level; if we observe certain results at the layer level, then at a subsection of that level we should observe the same results, unless knowledge is necessarily stored at the entire layer level. Since attention heads are being patched one at a time, the lack of attention-level hierarchies and variability in the p-values suggest that knowledge is stored across attention heads.

6 Discussion

While there is a clear trend of hierarchical storage in the results, the percentages of hyponyms being stored before a hypernym vary. We propose five key reasons why a hyponym may be stored *after* a hypernym across model-category pairs, testable in future work:

- *Distributed storage* - The concepts may be stored in multiple layers, introducing noise in the center-of-gravity metrics.
- *Depth* - The hyponym might be close in the semantic granularity to the hypernym.
- *Non-hierarchical storage* - The model may treat hierarchical concepts as distinct, unrelated concepts. When two unrelated concepts are allocated parameter storage spaces, it is equally likely that one appears in an earlier layer than the other.
- *Context* - The hyponym might have a different context than that of the hypernym in the training corpus of the language models.

In addition, the *frequency* of the hyponym might dictate whether it is stored after the hypernym (Wei et al., 2021). To assess this, we tested the association between frequency and hierarchical organization within model-category pairs using the Brown corpus. The tree category was excluded as most of its contents were not present in the corpus. 6/12 pairs reported a p-value of <0.05, indicating that there is a weak yet present association between frequency and structural hierarchies. Moreover, in the pairs that were significant, 4/6 had more frequent hyponyms when the hyponyms were stored after their hypernym. This may help explain why some model-category pairs have more pronounced hierarchies than others (e.g. the animal category, which had large p-values for Qwen-2.5-1.5B). Overall, we expect some noise due to distributive storage and context. However, the strength of our results indicates that multi-granular storage is present regardless. In addition, the tree category in Llama-2 had Wilcoxon and Binomial p-values of .999, .185 for the Binomial test in Mistral v0.1, and Wilcoxon and Binomial <0.001 p-values for the rest of the models (as shown in Table 2). We hypothesize the tree category is an outlier as it is a flatter hierarchy than that of the animal and food categories, which include several broad topics like bird/fish and dish/vegetables. Our hypotheses may explain the variability in results despite the fact that there are hierarchies present across model-category pairs.

Although the variability in the attention level and paired t-test results should indicate that concepts are stored across attention heads, we acknowledge that there are two competing hypotheses in this regard. If several attention heads encode a concept, then simply patching one of those heads should produce an effect on our logit difference metric and indicate that the head stores the concept. At the same time, patching a single attention head might not produce enough of an effect on the metric if the concept is distributed across multiple heads. Thus, there is a tug-of-war between a single attention head's effect and the distribution of the concept across multiple heads. Our results suggest that a single attention head does not produce enough of an effect on our metric; if it did, we would see a clear-cut trend in the t-test results. We also hypothesize that factual knowledge may be harder to locate in finer sections of the model, despite findings that this type of knowledge can be localized (Bahador, 2025). In addition, there are various competing mechanisms within LLMs that may cause concepts to be distributed across attention heads (and layers) (Ortu et al., 2024).

7 Conclusion

Our paper advances the field of MI by applying a causal method, activation patching, to multi-granular semantic hierarchies. Specifically, we use STR patching with a WordNet derived dataset to causally locate hierarchies of concepts within LLMs. We find that concepts at a finer level of granularity precede those at a coarser level in around 61-

78% of cases ($p < 0.01$); the hierarchical storage of these concepts is model- and category-specific. Further, the variability at the attention level indicates that concepts are stored across attention heads. These findings extend the work of Dalvi et al. (2022); Regneri et al. (2024); Tenney et al. (2019) by localizing conceptual abstraction and establishing semantic hierarchies at the layer and attention level. We strengthen the results of Aljaafari et al. (2024); Bronzini et al. (2024); Jawahar et al. (2019); Jin et al. (2025) by discovering how hierarchical concepts are organized within LLMs. Our findings go beyond just concept localization; we show that LLMs contain conceptual abstraction, where concepts are arranged according to semantic granularity. Ultimately, our work sheds light on abstraction within LLMs for more interpretable AI.

Limitations

While our approach provides evidence of hierarchies within LLMs, our methods present various limitations. Our dataset is liable to reliability errors (due to a single annotator) and leakage (due to LLM construction). In addition, we focus on 1.5-9B parameter models, and the generalizability of the results could be strengthened to models with different sizes (e.g., 70B, 135B) with more computing resources. Additionally, the patching of multiple attention heads would better explain the observed attention-level variability and offer insight regarding the correspondence between semantic and structural hierarchies. Further experiments could show why hyponyms are stored after hypernyms in 22-39% of cases. More importantly, additional research is needed to understand the dynamics of these hierarchies that emerge when the model is being trained.

References

Nura Aljaafari, Danilo S Carvalho, and André Freitas. 2024. The Mechanics of Conceptual Interpretation in GPT Models: Interpretative Insights. *arXiv preprint arXiv:2408.11827*.

Carlos Aspillaga, Marcelo Mendoza, and Alvaro Soto. 2021. Inspecting the concept knowledge graph encoded by modern language models. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2984–3000, Online. Association for Computational Linguistics.

Nooshin Bahador. 2025. Localized Definitions and Distributed Reasoning: A Proof-of-Concept Mechanistic Interpretability Study via Activation Patching. *ArXiv*, abs/2504.02976.

Marco Baroni and Alessandro Lenci. 2011. How We BLESSed Distributional Semantic Evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic Interpretability for AI Safety–A Review. *Transactions on Machine Learning Research*.

Marco Bronzini, Carlo Nicolini, Bruno Lepri, Jacopo Staiano, and Andrea Passerini. 2024. Unveiling llms: The evolution of latent representations in a dynamic knowledge graph. In *First Conference on Language Modeling*.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards Automated Circuit Discovery for Mechanistic Interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

W. J. Conover. 1999. *Practical Nonparametric Statistics*, third edition. edition. Wiley series in probability and statistics. Applied probability and statistics section. Wiley, New York ;.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2022. Discovering latent concepts learned in bert. In *International Conference on Learning Representations*.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2025. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. 2021. Causal Abstractions of Neural Networks. In *Advances in Neural Information Processing Systems*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning*, pages 15466–15490. PMLR.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks and Mantas Mazeika. 2022. X-Risk Analysis for AI Research. *arXiv preprint arXiv:2206.05862*.

M. Hollander, D.A. Wolfe, and E. Chicken. 2013. *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics. Wiley.

Guan Zhe Hong, Bhavya Vasudeva, Vatsal Sharan, Cyrus Rashtchian, Prabhakar Raghavan, and Rina Panigrahy. 2025. Latent Concept Disentanglement in Transformer-Based Language Models. *arXiv preprint arXiv:2506.16975*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *ArXiv*, abs/2310.06825.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573, Abu Dhabi, UAE. Association for Computational Linguistics.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. *arXiv preprint arXiv:2307.09458*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. *Advances in neural information processing systems*, 35:17359–17372.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. 2024. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8420–8436, Bangkok, Thailand. Association for Computational Linguistics.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The Geometry of Categorical and Hierarchical Concepts in Large Language Models. In *ICLR*.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Michaela Regneri, Alhassan Abdelhalim, and Soeren Laue. 2024. Detecting conceptual abstraction in LLMs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4697–4704, Torino, Italia. ELRA and ICCL.

Eleanor Rosch, Carolyn Mervis, Wayne Gray, David Johnson, and Penny Braem. 1976. Basic Objects in Natural Categories. *Cognitive Psychology - COG PSYCHOL*, 8:382–439.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, and Alexandre Ramé. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint arXiv:2408.00118*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4593–4601.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. *arXiv preprint arXiv:2004.12265*.

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

Zijian Wang, Britney Whyte, and Chang Xu. 2024. Locating and extracting relational concepts in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4818–4832, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics bulletin*, 1(6):80–83.

xAI. 2024. Grok-3: A Large Language Model. Accessed: October 2025.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. Qwen2.5 Technical Report. *ArXiv*, abs/2412.15115.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

A Additional Results

To test whether hierarchical concepts are stored at the attention level before the layer level, we paired each concept's layer and attention level center-of-gravity for all models and categories. We then conducted a paired t-test to determine if there was a difference between each concept's attention and layer metric. If the p-value was close to zero, then the tests suggest that there is a difference between the Centers of Gravity. The p-values are at opposite extremes across model-category pairs (see Table 3), which we hypothesize is a consequence of the variability in the attention level results. This is also evident in Figure 3, where the percentages of hyponyms being stored before hypernyms varies significantly. Overall, this suggests that knowledge is distributed across attention heads.

Model	Category	Paired t-test p-value
Llama-2-7B	Animal	0.000*
	Food	1.000
	Tree	0.000*
Mistral v0.1	Animal	0.000*
	Food	1.000
	Tree	0.001*
Gemma-2-9B	Animal	1.000
	Food	1.000
	Tree	1.000
Llama-3.2-3B	Animal	0.000*
	Food	1.000
	Tree	0.000*
Gemma-2-2B	Animal	1.000
	Food	1.000
	Tree	1.000
Qwen-2.5-1.5B	Animal	0.000*
	Food	0.001*
	Tree	1.000

Table 3: Paired t-test to compare attention and layer indices. Significant results with p-values < 0.05 are marked with $*$.

B Layer Center-of-Gravity Ablation

We conducted an ablation study to determine whether k = 5 for the top-k layer center-of-gravity was robust. We recalculated the layer metrics for each concept, where k = 1, 3, 7, 10, all, and found the Spearman Rank correlation between each of the top-k metrics and the top-5 metric on a holdout

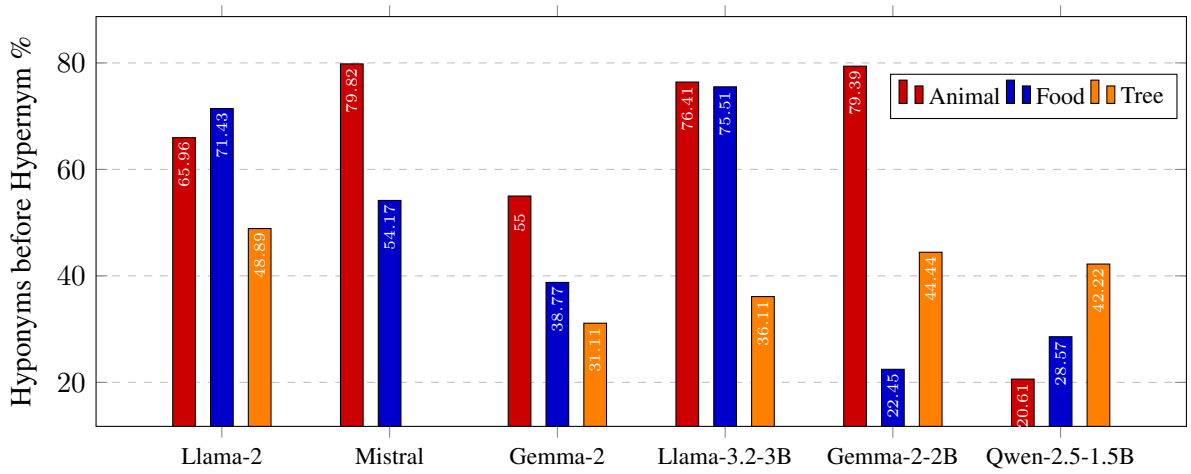

Figure 3: Attention Level: Percentage of hyponyms being stored before hypernyms across models and categories. The percentage for the Mistral-Tree pair was 0.0%.

set of concepts (20%) followed by the remaining set (80%). Since k = 3, 7, 10 has high correlations (>.9) with k = 5 across models and categories, the layer metric is less sensitive to changes in k near k = 5 and therefore the top-5 metric is robust (see Table 4). This was confirmed in both sets. The only exception is the tree category for the Llama-2 model, which also performed poorly in the statistic tests. This was considered an outlier, and we maintained k = 5 as it is robust and optimal for nearly all categories and models.

C Attention Center-of-Gravity Ablation

We conducted another study to provide evidence of robustness for m = 3 in $m\sigma$ as part of the calculation of the attention metric. We varied m = 2, 2.5, 3.5, 4 and recalculated the Centers of Gravity for every concept. We then found the Spearman Rank correlation between each of the recalculated Centers of Gravity and the m = 3 Centers of Gravity on a holdout set of concepts (20%) followed by the remaining set (80%). Since m = 2.5, 3.5 has high correlations (>.7) with m = 3, the attention metric is less sensitive to changes near m = 3 and therefore the attention center-of-gravity is robust (see Table 4). This was confirmed in both sets. The only exception is the tree category for Mistral v0.1, which had too little attention head occurrences across the multipliers to provide valid correlations. We argue that this is a consequence of single head patching (as opposed to multi-head patching) rather than a fault in our methodology.

D Prompts for Dataset Generation

The following example prompts were used to generate the dataset: Step 1: I want the "sampled-food-nouns.txt" to be sorted into a hierarchy. All 300 nouns in the file should be used. Create a dictionary where each key is a topic such as food or nutrient. The values should be a list of elements that are hyponyms of the key noun. These elements should all be at one level of granularity below the key. Thus, the elements of a values list are all at similar levels of granularity. Further, the elements of a values list should appear as keys later on in the dictionary if the elements also have hyponyms that can be stored as a list. There should not be any keys with empty lists as values. Please create this dictionary manually. The values can only be taken from the file, but the keys can be taken from the WordNet 3.1 database. Ensure that all keys are within the database. Make sure there are no duplicate elements within the values list. If there are duplicate elements, delete the ones whose keys are at the higher levels of granularity.

Step 2: I am going to use these datasets for a "patching" experiment with LLama. In the experiment, I will prompt Llama with a source prompt; this source prompt will result in a specific hidden representation within the model. For example, a source prompt could be "A dish made by frying scrambled eggs in a pan is called omellete". This prompt corresponds with omelette; the hidden representation of omelette will then be patched into a target prompt. The target prompt should be a prompt that reveals a property about a topic different than that of the source prompt. For example,

Model	Category	Layer CoG Top-K vs. Top-5					Attention CoG $m\sigma$ vs. 3σ			
		k = 1	k = 3	k = 7	k = 10	k = all	m = 2	m = 2.5	m = 3.5	m = 4
Llama-2-7B	Animal	.920	.977	.991	.981	.936	.709	.874	.817	.733
	Food	.896	.967	.980	.965	.867	.700	.844	.818	.604
	Tree	.584	.682	.723	.642	.420	.579	.674	.663	.535
Mistral v0.1	Animal	.917	.970	.987	.968	.852	.794	.909	.952	.852
	Food	.900	.979	.980	.953	.874	.712	.881	.845	.711
	Tree	.799	.911	.971	.961	.676	N/A	N/A	N/A	N/A
Gemma-2	Animal	.924	.976	.989	.965	.885	.848	.934	.911	.838
	Food	.763	.948	.923	.876	.678	.835	.877	.881	.790
	Tree	.756	.946	.950	.942	.717	.704	.784	.833	.638
Llama-3.2-3B	Animal	.851	.959	.986	.958	.770	.794	.895	.813	.721
	Food	.882	.982	.973	.918	.821	.556	.659	.893	.789
	Tree	.844	.910	.957	.952	.711	.871	.816	.905	.653
Gemma-2-2B	Animal	.893	.981	.987	.965	.900	.870	.940	.921	.886
	Food	.727	.890	.936	.913	.698	.808	.808	.999	.926
	Tree	.717	.948	.966	.913	.563	.490	.615	.923	.847
Qwen-2.5-1.5B	Animal	.864	.965	.978	.943	.885	.633	.846	.913	.731
	Food	.809	.981	.981	.964	.880	.860	.936	.932	.898
	Tree	.861	.959	.960	.970	.923	.720	.901	.887	.801

Table 4: Spearman Rank Correlations for layer center-of-gravity (Top-K vs. Top-5) and attention center-of-gravity ($m\sigma$ vs. 3σ).

the target prompt that corresponds with the source prompt above could be "the color of an apple is...". Thus, when the hidden representation of the answer of the source prompt is patched into the target prompt, the answer should be yellow. I want you to create two dictionaries within the same JSON file. The first is the "prompts" dictionary, which should contain a "source" and "target" key for each key within the food-dataset file. For example, since "food" is a key within the food-dataset file, there should be a "food-source" and "food-target" key within the prompts dictionary. Each new key should have a list of prompts. For each element within the values list of the corresponding key in the food-dataset file, the "source" key should have a prompt who states the answer, and the "target" key should have a corresponding prompt about a different subtopic such that when the source hidden representation is patched into the target prompt, the prompt will result in an answer different than that of the original target prompt. In other words, the target prompt answer and patched answer should not be the same. The prompts at the same index of the values list of corresponding "source" and "target" keys will be paired together for the patching experiment. I also want you to create an "answers" dictionary. This dictionary should have identical keys to that of the food-dataset file that correspond to the "source" and "target" keys from the prompts dictionary. The values should be a list of tuples where the first index is the subtopic of the prompt, the second index is the token in the target prompt that is being patched into, and the third index is the answer that occurs when the source hidden representation is patched into its paired target prompt.

Note that Grok-3 outputs may vary and some instructions may need to be repeated in order to obtain the dataset.