Analogical Reasoning Inside Large Language Models : Concept Vectors and the Limits of Abstraction

Anonymous ACL submission

Abstract

Analogical reasoning relies on conceptual abstractions, but it is unclear whether LLMs harbor such internal representations. We explore distilled representations from LLM activations and find that function vectors ($\mathcal{FV}$s; Todd et al., 2024)—compact representations for in-context learning (ICL) tasks—are not invariant to simple input changes (e.g., open-ended vs. multiple-choice), suggesting they capture more than pure concepts. Using representational similarity analysis (RSA), we localize a small set of attention heads that encode invariant concept vectors ($\mathcal{CV}$s) for verbal concepts like *antonym*. These $\mathcal{CV}$s function as feature detectors that operate independently of the final output—meaning that a model may form a correct internal representation yet still produce an incorrect output. Furthermore, $\mathcal{CV}$s can be used to causally guide model behaviour. However, for more abstract concepts like *previous* and *next*, we do not observe invariant linear representations, a finding we link to generalizability issues LLMs display within these domains.

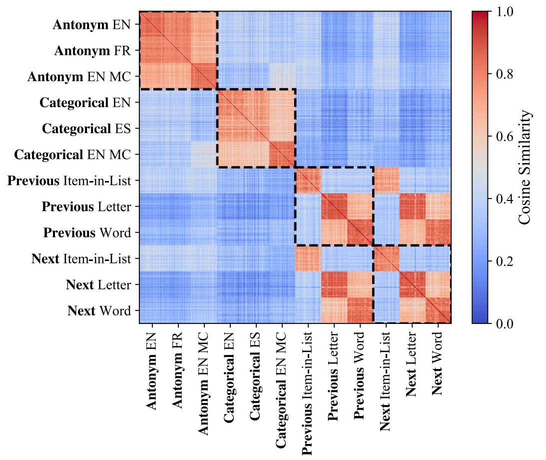

Figure 1: Pairwise similarity matrix of $\mathcal{CV}$'s extracted from Llama-3.1 70B across 600 ICL prompts covering various concepts and low-level presentations. $\mathcal{CV}$'s remain invariant for the verbal concepts *antonym* and *category*, but show no stable representation of abstract concepts like *previous* or *next*. Instead, these tasks exhibit order-based representations tied to known lists (e.g., alphabets, weekdays) or low-level clustering based on presentation format (words vs. letters).

1 Introduction

"Analogies are functions of the mind" (Hill et al., 2019, p.10). People use analogies to flexibly map previous knowledge to novel domains (Hofstader, 1979; Mitchell, 2020). For example, if you are just beginning to learn about analogical reasoning, envisioning a "bridge" that connects new information to concepts you already understand can be very helpful. In essence, successful analogy-making depends on our ability to extract and apply conceptual abstractions—such as "bridge" or "connection"—from seemingly unrelated situations. While behavioral evidence suggests that analogical reasoning have emerged in LLMs (Brown et al., 2020; Webb et al., 2023), it remains unclear if and how LLMs represent these relational concepts internally.

What does it mean for a neural system to represent abstract concepts? We formalize abstraction as *conceptual invariance*.

Consider a high-level concept $\mathcal{C}$ (e.g., "antonym"). A neural network f flexibly represents $\mathcal{C}$ if it encodes the same abstract representation regardless of variations in its low-level inputs. Let X denote the space of all inputs encoding $\mathcal{C}$ and $\mathcal{T}$ a group of transformations on X (e.g., changes in language, format, or modality) that preserve the concept's meaning. Then, f satisfies conceptual invariance if

$$f(t(x)) = f(x), \quad \forall t \in \mathcal{T}.$$

This ensures that the network's encoding of C reflects its essence rather than superficial characteristics of low-level input. This is analogous to how object representations in convolutional neural networks are translation-invariant (Lecun et al., 1998).

Previous Work Previous work identified *Function Vectors* ($\mathcal{FV}$s; Todd et al., 2024; Hendel et al., 2023), a compact vector representation of an ICL task (Brown et al., 2020). The representation is encoded by a universal set of attention heads (high overlap of heads across different tasks), and can be transplanted into the model internals to causally guide its behavior (even zero-shot - e.g. transplanting an antonym $\mathcal{FV}$ to a prompt 'fast: ' induces the network output 'slow'). Attention heads composing the $\mathcal{FV}$ were found using activation patching a popular mechanistic interpretability technique for localizing information in neural networks (Heimersheim and Nanda, 2024; Details in Section 2.6).

Summary of contributions We investigate whether conceptual invariance holds for $\mathcal{FV}$s and find they are not invariant to low-level changes (e.g., switching the ICL format from open-ended to multiple-choice; Section 3.1). Instead $\mathcal{FV}$s encode dense, detailed information that goes beyond the latent conceptual content we were targeting (Section 3.2). Based on additional checks we conclude that activation patching itself may be responsible for this shortcoming, as it appears to overlook the true latent representations (Section 3.3).

We then use representational similarity analysis (RSA; 2.8) to localize latent abstract information in transformer internals. For verbal concepts, we find a set of attention heads emerging in early-to-mid layers (Section 4.1). By summing their outputs we form the $\mathcal{CV}$. We find that the extent of conceptual invariance grows with number of training examples in the ICL prompts. Interestingly, we find that $\mathcal{CV}$s can carry the correct conceptual representations while the model produces incorrect answers (Section 4.2). We then ask whether the $\mathcal{CV}$s causally influence behaviour 4.3. We find that while being much weaker at zero-shot interventions, with enough context in the prompt, $\mathcal{CV}$s influence model output and do so in a more portable manner than $\mathcal{FV}$s.

Finally, we use $\mathcal{CV}$s to demonstrate that our LLMs did not develop representations of abstract concepts of 'Previous' and 'Next' (Figure 1). We further use our findings to inform the discussion of analogical reasoning capabilities in LLMs through the lens of internal model representations (Section 6).

2 Materials and Methods

2.1 Models

We investigate the LLama 3.1 model family (Grattafiori et al., 2024), specifically on the 8 and 70 billion parameter variants.

Llamas are autoregressive, residual-based transformers. The models, f internally comprise of $\mathcal{L}$ layers. Each layer is composed of a multi-layer perceptron (MLP) and J attention heads $a_{\ell j}$ which together produce the vector representation of the last token, $\mathbf{h}_\ell = \mathbf{h}_{\ell-1} + \text{MLP}_\ell + \sum_{j \in J} a_{\ell j}$ (Elhage et al., 2021). In all our experiments we focus on the representations extracted from the last token position.

2.2 Task Formulation

For every dataset $d \in D$ in our collection, we define a set P_d containing in-context prompts $p_d^i \in P_d$.

Each prompt p_d^i is a token sequence that includes N input-output exemplar pairs (x, y), all illustrating the same underlying concept $\mathcal{C}$ and its corresponding mapping from x to y. Additionally, each prompt provides a query input x_q^i linked to a target response y_q^i. y_q^i is not shown to the model and we consider that the model performs correctly on p_d^i if its predicted token matches y_q^i (or the first token of y_q^i for multi-token words).

2.3 Verbal Concepts

Translation We use English-to-French and German-to-Spanish tasks.

Antonym We source antonym word pairs from Todd et al. (2024). E.g.,: `Big` $\rightarrow$ `Small`.

Categorical We generate 1000 pairs using OpenAI's GPT-4o. E.g.,: `Table` $\rightarrow$ `Furniture`.

Low-level transformations We test verbal concepts in three low-level presentations - Open-ended in English, Open-ended in a different language, and Multiple-Choice (MC) in English.

2.4 Abstract Concepts

We investigate two abstract concepts, **Previous** and **Next**, capturing whether an entity comes before or after another entity. We test these concepts using three different low-level presentations:

Item in List Our pairs are made up of days of the week, months of the year, letters of the alphabet, and number pairs (both numeric and text form). Some examples for Next-Item in List: `Monday` $\rightarrow$

Concept	Dataset	Question Type	Response Type	Info Source	Lang
Translation	English to French	open	word	not in prompt	FR
	German to Spanish	open	word	not in prompt	ES
	English to French-MC	MC	letter	in prompt	-
Antonym	Antonym EN	open	word	not in prompt	EN
	Antonym FR	open	word	not in prompt	FR
	Antonym MC	MC	letter	in prompt	-
Categorical	Categorical EN	open	word	not in prompt	EN
	Categorical ES	open	word	not in prompt	ES
	Categorical MC	MC	letter	in prompt	-
Previous	Prev Item-in-List	open	mixed	not in prompt	-
	Prev Abstract-Letter	open	letter	in prompt	-
	Prev Abstract-Word	open	word	in prompt	EN
Next	Next Item-in-List	open	mixed	not in prompt	-
	Next Abstract-Letter	open	letter	in prompt	-
	Next Abstract-Word	open	word	in prompt	EN

Table 1: Task Information Table

`Tuesday`, `December` $\rightarrow$ `January`, `a` $\rightarrow$ `b`, `seven` $\rightarrow$ `eight`.

And for Previous-Item in List: `Tuesday` $\rightarrow$ `Monday`, `January` $\rightarrow$ `December`, `a` $\rightarrow$ `z`, `eight` $\rightarrow$ `seven`.

Abstract Previous/Next Task We evaluate tasks where a sequence contains one indicator element, one target element, m distractors sharing the target's features, and n positional elements that do not. The target always appears either before (Previous) or after (Next) the indicator. We test two variants—using either English words or letters (a, b, c, d)—with one-token elements. Below we show examples for $m = 3$, $n = 3$ with indicator elements being "`*`" and positional "`.`". The target elements are "`c`" and "`letter`".

Previous-Letter Example:

```
Q: . a c . * b . d   A: c
Q: c a * . . d b .   A: a
Q: b a d c . . *     A:
```

Next-Word Example:

```
Q: . big mask . * control . house
A: control
Q: star code * . . dense light .
A: dense
Q: ball might poland * . letter .
A:
```

2.5 Task Attributes

Our tasks have high-level (concepts) and low-level attributes: **Question Type** - ICL prompt in either open-ended or multiple-choice (MC) format; **Response Type** - whether the expected response is a word, letter, or a mix of both; **Information Source** - whether the expected response is located somewhere in the prompt (e.g., MC items), or needs to be generated (e.g., open-; **Language**-the language of the expected response.

2.6 Activation Patching

Activation patching replaces specific activations with cached ones from a *clean* run to assess their impact on the model's output. The cached activations are then inserted into selected model components in a *corrupted* run, where the systematic relationships in the prompt are disrupted. For example, in an antonym ICL task, consider a *clean prompt*:

```
Hot -> Cold : Big -> Small : Clean -> ?
```

and a *corrupted prompt*:

```
House -> Cold : Eagle -> Small : Clean -> ?
```

To localize attention heads carrying task-relevant information we compute the *causal indirect effect* (CIE) for each attention head $a_{\ell j}$ as the difference between the probability of predicting the expected answer y when processing the corrupted prompt $\tilde{p}$ with and without the transplanted mean activation $\bar{a}_{\ell j}$ from clean runs:

$$\mathrm{CIE}\big(a_{\ell j}\big) = f\Big(\tilde{p} \mid a_{\ell j} := \bar{a}_{\ell j}\Big)[y] - f\big(\tilde{p}\big)[y].$$

We then compute the *average indirect effect* (AIE) over a collection $\mathcal{D}$ of 10 datasets from Todd et al. (2024):

$$\mathrm{AIE}(a_{\ell j}) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{1}{|\tilde{\mathcal{P}}_d|} \sum_{\tilde{p}_i \in \tilde{\mathcal{P}}_d} \mathrm{CIE}\big(a_{\ell j}\big),$$

where $\tilde{\mathcal{P}}_d$ denotes the set of corrupted prompts for dataset d.

2.7 Function Vectors

A function vector for a specific dataset ($\mathcal{FV}_d$) is computed as the sum of the mean activations over all clean prompts from the dataset from a set $\mathcal{A}_{\mathcal{FV}}$ of top N attention heads having the highest AIE values:

$$\mathcal{FV} = \sum_{a_{\ell,h} \in \mathcal{A}} \bar{a}_{\ell,h}.$$

Following the implementation in Todd et al. (2024), we set $N = 20$ for the 8B model and $N = 100$ for the 70B model.

2.8 Representational Similarity Analysis

To distill conceptual information from LLMs during ICL, we employ representational similarity analysis (RSA)—a technique invented for cognitive neuroscience (Kriegeskorte, 2008). In our work, RSA is used to assess the alignment between LLM representations and task attributes.

For each $a_{\ell j}$ we compute representational similarity matrices (RSMs) of the form:

$$\text{RSM} = \begin{bmatrix} 1 & \cdots & \theta(v_1, v_N) \\ \vdots & \ddots & \vdots \\ \theta(v_N, v_1) & \cdots & 1 \end{bmatrix}$$

where v_i denotes the output extracted from $a_{\ell j}$ for the ith prompt $p_i \in P_N$, and $\theta(\cdot,\cdot)$ is a similarity function.

Additionally, for each task attribute q (i.e., `concept`, `info_source`, `lang`, `response_type`, `task_type`), we construct $N \times N$ binary design matrix DM_q, where each entry is set to 1 if the corresponding pair of prompts share the same attribute value, and 0 otherwise.

We then quantify the alignment between the lower-triangular portions of the RSM and DM_q using the non-parametric Spearman's rank correlation coefficient. This alignment for $a_{\ell j}$ is denoted by $\Phi^q_{\ell j}$. When referring to $\Phi^{\texttt{concept}}$ we mean the alignment between model activations and the subset of datasets containing *verbal* concepts only, unless stated otherwise.

2.9 Concept Vectors

Analogous to $\mathcal{FV}$s (Section 2.7), the ($\mathcal{CV}_d$)'s are constructed by summing the mean activations from a set of top-ranking attention heads. In this case, we sum the top 3 attention heads with the highest $\Phi^{\texttt{concept}}$ scores, forming a set $\mathcal{A}_{\mathcal{CV}}$, for both model sizes.

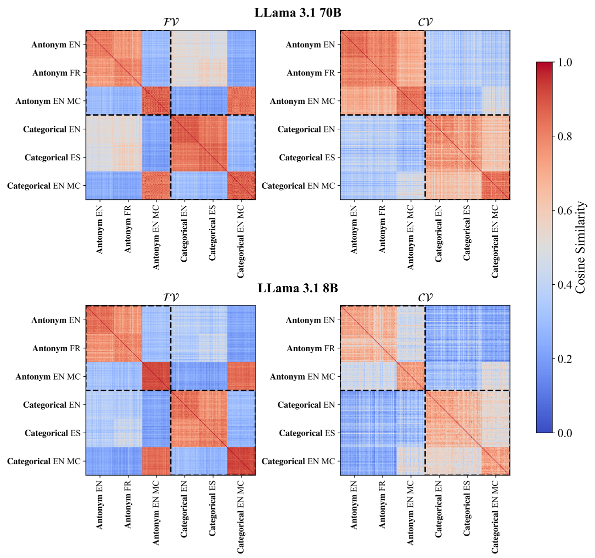

Figure 2: Representational similarity matrices for antonym and categorical concepts each tested with three low-level transformations. The upper-left and lower-right quadrants (outlined with the dashed lines) contain pairwise similarity scores for prompts coming from the same concept. $\mathcal{CV}$s encode the concept in a more invariant manner than $\mathcal{FV}$s.

3 Do $\mathcal{FV}$s create an invariant representation of latent concepts?

We start our search for invariant conceptual representations using methods that rely on activation patching. We show that $\mathcal{FV}$s carry more than purely relational information, and that diversifying the datasets does not help localize the attention heads carrying latent information.

3.1 $\mathcal{FV}$s are not invariant to low-level transformations

We extract $\mathcal{FV}$s per prompt for all of the datasets outlined in 1. That is for prompt $i \in N$ prompts $\mathcal{FV}_i = \sum a^i_{\ell j}$, where $a_{\ell j} \in A_{\mathcal{FV}}$. Each dataset had 50 prompts, each consisting of a 5-shot ICL task.

As we see in Figure 2 $\mathcal{FV}$ representations cluster within the concepts in both languages in open-ended question formats, but the clustering disappears for multiple-choice prompts, where all items cluster together, despite encompassing multiple concepts (e.g., antonym and categorical MC items show high similarity - they are represented using a subspace that is orthogonal to open-ended items). This shows that $\mathcal{FV}$ representations are contextual rather than conceptually invariant.

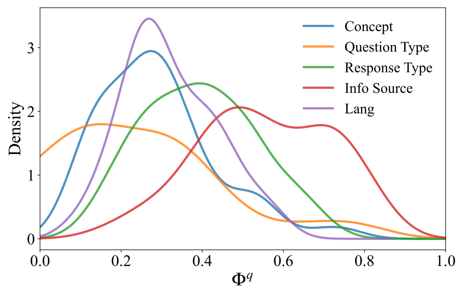

Figure 3: Density plot displaying the information-rich make-up of 100 attention heads in LLama 70B comprising its $\mathcal{FV}$.

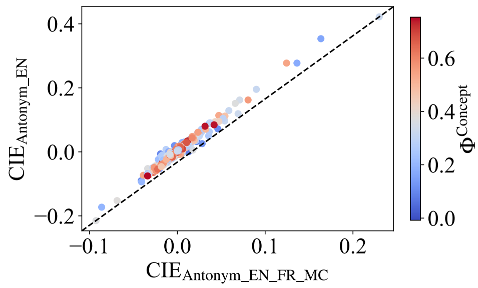

Figure 4: Patching activations from multiple low-level manifestations of a latent concept does not change which attention heads are ranked to have the highest causal effect nor does it help localize latent conceptual information.

3.2 $\mathcal{FV}$s encode multiple task attributes

This leads us to the question *what* information $\mathcal{FV}$s encode, if not purely the concepts? We answer this by investigating how much each task attribute explains the activation spaces of each attention head in $A_{\mathcal{FV}}$.

Figure 3 displays density plots for all $\Phi^q_{\ell j}$. These plots reveal that each task attribute is represented to some extent within the $\mathcal{FV}$s, with `task_type` exhibiting the highest density. This indicates that the attention heads forming the $\mathcal{FV}$s are particularly sensitive to whether the language model is tasked with extracting information from the input prompt or generating a novel token. This sensitivity aligns with the RSM shown in Figure 2—multiple-choice items form distinct clusters because they are extractive (in contrast to open-ended items) and have a different response type (four possible letters versus words). Importantly, while relational information is present, it does not play a crucial role in shaping the $\mathcal{FV}$s, confirming that $\mathcal{FV}$s are not invariant representations of latent concepts.

3.3 Activation Patching Does Not Localize Latent Components

Attention heads in the $\mathcal{FV}$s were identified using activation patching on a single low-level manifestation (e.g., English antonyms). To test whether the failure to localize latent conceptual information is due to data selection or the method itself, we computed the CIE for all attention heads for antonyms across three manifestations ($\text{CIE}_{\text{antonym_eng_fr_mc}}$) and compared it to $\text{CIE}_{\text{antonym\_eng}}$.

The top 100 heads ranked by both metrics overlap by 89%, indicating that adding more low-level datasets does not significantly change the $\mathcal{FV}$ composition. One might argue that choosing 100 heads is somewhat arbitrary and that varying this number could potentially highlight relational information more effectively. To investigate this possibility, we examined the raw CIE values for each dataset composition. As shown in Figure 4, there is a strong correlation between $\text{CIE}_{\text{antonym\_eng}}$ and $\text{CIE}_{\text{antonym\_eng\_fr\_mc}}$. In other words, adding more low-level prompts does not alter which attention heads are ranked as having higher causal importance in producing the expected output.

Finally, we note that many attention heads with high Φ^{concept} scores are scored low by the CIE metrics, demonstrating that activation patching is not effective at identifying latent components. More broadly, since activation patching can localize causal, but not latent components, it implies that latent information plays only a small role in next-token prediction (much like knowing an answer to a multiple-choice exam but not the "abcd" response format).

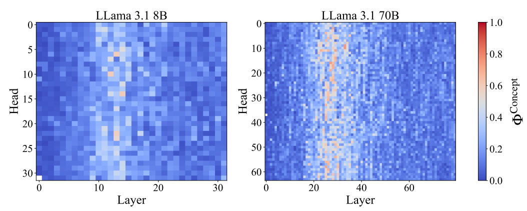

Figure 5: Attention heads encoding verbal concepts emerge in early-to-mid layers.

4 $\mathcal{CV}$s emerge for verbal concepts

In order to distill invariant conceptual representations in LLMs we turn to RSA (Sec. 2.8). In this section we report on our findings regarding $\mathcal{CV}$s.

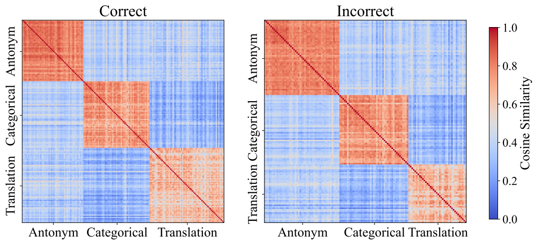

Figure 6: Concept representation can be independent from the model's output. $\mathcal{CV}$s can encode the correct concept while the model produces the incorrect response. *Note*: we do not show multiple-choice items as performance was too high ($> 90\%$) to contrast correct (N =168) vs incorrect activations (N =132).

4.1 $\mathcal{CV}$s are invariant to low-level transformations

Our analysis reveals strong clusters in the $\mathcal{CV}$ representational space that are delineated by verbal concepts (Figure 2). Compared to the $\mathcal{FV}$s, the $\mathcal{CV}$ representations are more *invariant* to low-level transformations and more *specific*—that is, pairwise similarities between different concepts are lower than those within the same concept. While there is a high similarity (Mean = 0.8) among items of the same concept in different languages, the mean similarity drops to 0.7 when items are presented as MC format instead of open-ended. This shows that $\mathcal{CV}$s, while being close to our notion of conceptual invariance, are not perfect.

4.2 $\mathcal{CV}$s are feature detectors

Figure 9 shows that model accuracy improves with Φ_{concept} as the number of training examples N increases, suggesting that the ability to form invariant representations of the underlying concepts is linked to task performance. However, as illustrated in Figure 4.2, the model sometimes forms accurate $\mathcal{CV}$s even when it predicts the incorrect answer. We interpret this as evidence that the model employs $\mathcal{CV}$s as feature detectors. This finding points to a mechanism where the model identifies latent concepts in its early-to-mid layers (see Figure 5), which may then, or may not, be leveraged in later layers to predict the next token. In cases where the model selects an incorrect token, it may be due either to uncertainty about the specific item or because the correct answer is ambiguous.

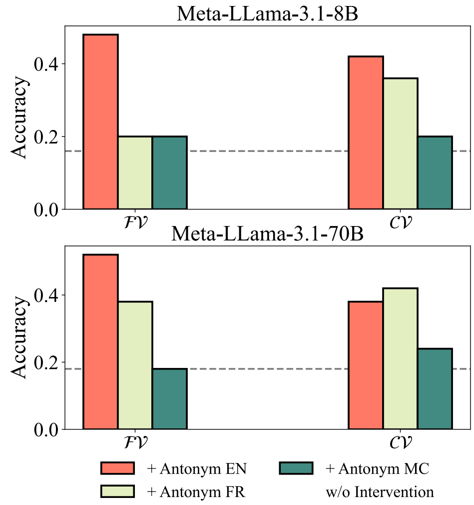

Figure 7: The effect of adding $\mathcal{CV}$s and $\mathcal{FV}$s extracted from in-distribution (Antonym EN) and out-of-distribution (Antonym FR and Antonym MC) prompts to the models' hidden states when performing *AmbigousICL*. The grey dashed line shows baseline performance without intervention. $\mathcal{CV}$s causally guide behaviour model behaviour and are more portable than $\mathcal{FV}$s.

4.3 $\mathcal{CV}$s can causally guide model's behavior

As we showed, $\mathcal{CV}$s selectively and invariantly represent verbal concepts, even when the final behavior of the model is incorrect. This raises the question whether the model even uses the information encoded by $\mathcal{CV}$s. Using causal interventions, and an adapted task we call *AmbigousICL* we show that yes, the models use $\mathcal{CV}$s.

AmbigousICL We create a task where we randomly interleave two different ICL concepts in the training examples.

AmbiguousICL Example:

```
Q: indoor      A: outdoor
Q: noise       A: bruit
Q: western     A: eastern
Q: add         A: ajouter
Q: abstract    A: abstrait
Q: export      A:
```

We intervene with $\mathcal{CV}$s by adding them to hidden states at different layers, $\mathbf{h}_\ell$, while the model processes a 10-shot *AmbigousICL* prompt and then measure model performance in task execution. We find the best layer to intervene by testing the performance on *AmbigousICL* with the $\mathcal{CV}$s extracted from 50 prompts in the Antonym EN task. We found these to be layers 14 and 31 for 8B and 70B models respectively (roughly corresponding to where the attention heads encoding verbal concepts emerge, see Figure 5). For $\mathcal{FV}$s we follow Todd

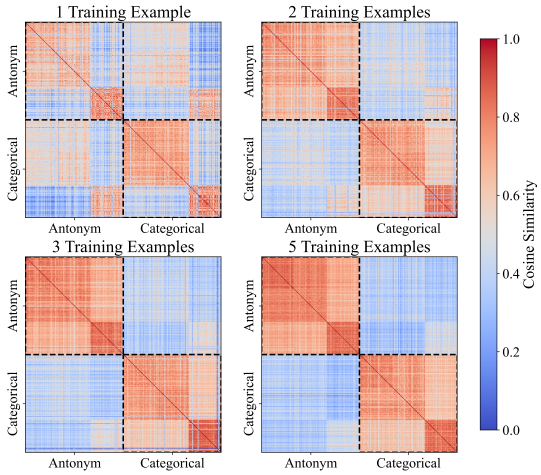

Figure 8: Representational invariance in Llama 3.1 70B grows with the number of training examples in the ICL prompt. The biggest difference is visible from 1 to 2 training examples where $\mathcal{CV}$s, similarly to $\mathcal{FV}$s in Figure 2), first cluster according to low-level similarity and then display a more invariant representational space, similar to the one in 5 training examples.

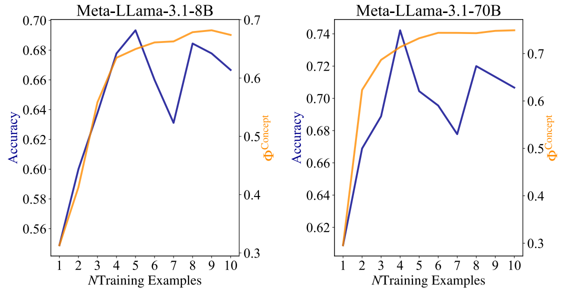

Figure 9: Φ_{concept} grows hand-in-hand with mean accuracy as a function of N training examples in the ICL prompt, while $N < 5$, and then plateaus. *Note*: Error bars around accuracies were removed to reduce clutter.

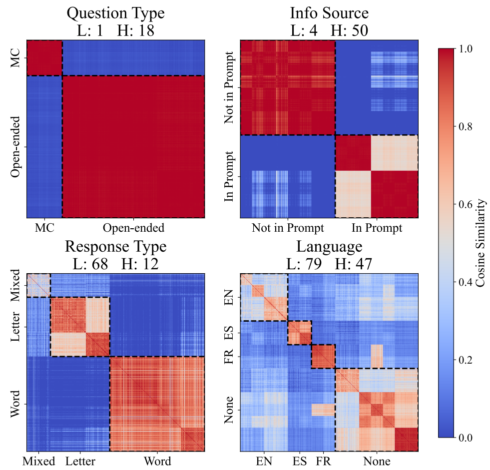

Figure 10: Attention heads with the highest Φ^q for each task attribute, q. Info source and Question Type emerge early in the transformer, while Language and Response Type in late layers.

et al. (2024) recommendation and use the third of the total layer count. We find that $\mathcal{CV}$s work best if you apply 10x scaling and 1x for $\mathcal{FV}$s.

We test both the causal power and the portability of the distilled representations. We extract $\mathcal{FV}$s and $\mathcal{CV}$s from three low-level manifestations of the concept Antonym (open-ended EN, open-ended FR, and MC) and transplant them inside of the models while they process the *AmbigousICL* task.

We find that intervening with $\mathcal{CV}$s increases the probability of model returning the antonym continuation. While $\mathcal{FV}$s are more effective at guiding the model behaviour when extracted from the same distribution of the task (open-ended EN antonym), they perform worse than $\mathcal{CV}$s when extracted from Antonym FR (even though $\mathcal{CV}$s are constructed from a much smaller number of attention heads than $\mathcal{FV}$s).

However, when extracting from MC items, performance reduces almost to baseline for both $\mathcal{CV}$s and $\mathcal{FV}$s. This provides interesting information regarding how similar vector representations should be in order to achieve similar intervention performance. In case of $\mathcal{CV}$s the mean similarity of 0.8 between Antonym EN and Antonym FR tasks is enough to achieve the same performance while the similarity of 0.7 between Antonym EN and Antonym MC is not.

Finally, in a zero shot setting $\mathcal{FV}$s work much better than $\mathcal{CV}$s (50% vs. 14% for Llama 8b and 58% vs. 2% for Llama 70b). Overall, these results suggest that $\mathcal{CV}$s capture purer latent conceptual representations, while $\mathcal{FV}$s also embed lower-level task details that are necessary for correct output.

5 Our method also localizes other task attributes

While this paper focuses on conceptual information in LLMs, we find that using RSA is also fruitful to localize model components where representational spaces align with other task attributes (Figure 10).

6 Lack of Abstract Concept Representations Impedes Generalization

Figure 1 shows that abstract concepts are not encoded as linear representations in $\mathcal{CV}$s. We find no attention heads with $\Phi^{\text{concept\_abstract}}$ scores exceeding 0.16 (compared to a maximum $\Phi^{\text{concept\_verbal}}$ of 0.75), confirming that abstract representations do not emerge elsewhere in the model.

However, task performance is high (the 70B model achieves 98% accuracy for previous/next items and 62% for abstract previous/next tasks). This implies that LLMs rely on alternative strategies rather than using explicit, top-down representations of abstract concepts such as "Previous" and "Next". One might ask: if the models perform well without abstract representations, what is the drawback? We now show that without reusable abstract concepts, models struggle to generalize to new domains.

Letter-string Tasks Hofstader (1979) introduced letter-string analogies to study human analogy-making in a simplified domain. These tasks require understanding “Next” and “Previous” concepts (e.g., given the normal alphabet, if “abc” becomes “abd”, then “ghi” should become “ghj”). Lewis and Mitchell (2024) found that GPT-4’s performance degrades as the alphabet deviates from its canonical order (e.g., “a b c e d f ...” is easier than “f e b a d c ...”), suggesting that it uses memorization rather than abstraction to solve the task.

We adopt the prompts from Lewis and Mitchell (2024), extracting $\mathcal{CV}$s from 20 prompts per alphabet (covering five permuted Latin alphabets and one symbolic alphabet such as “# $ * ! @”). Each prompt shows the alphabet with a one-shot ICL example (adapted for non-instruction tuned models). Because Llama 3.1 70B yielded near-zero accuracy on “previous” items, we focus solely on the “next” concept. We also extract $\mathcal{CV}$s from our "Next Item-in-List" and "Next Abstract-Letter" items (see Section 2.4).

N_{perm}	0	2	5	10	20	Symb
Accuracy	0.35	0.10	0.05	0.00	0.00	0.15

Table 2: Accuracy in LLama-3.1 70B goes down on Letter-String tasks the more the alphabet deviates from the memorized one (N_{perm}=0). The chance level is 0.04 for the letter alphabets and 0.1 for the symbol alphabet.

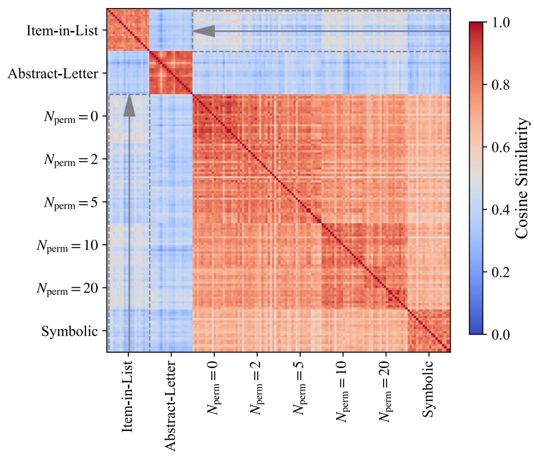

Figure 11: RSM of $\mathcal{CV}$s extracted from LLama-3.1 70B when performing Letter-String tasks with N permutations, and other tasks with the concept "Next". The arrows show what the gradient of similarities would look like if the $\mathcal{CV}$s had a shared representation of ordered lists.

Consistent with our findings so far, we do not see an invariant representation of the concept "Next" across the tasks (Figure 11). Instead, each task forms its own distinct cluster. Surprisingly, this also suggests that the model represents memorized lists differently in the Next Item-in-List and Letter String tasks. If these representations were shared, we would expect to see a gradient of similarities that decreases with increased alphabet shuffling. This absence might be due to differences between the tasks—for example, the inclusion of the alphabet in the Letter-String prompts or the presence of additional memorized lists in the Next Item-in-List task. In any case, these findings highlight that the model’s representations are highly contextual on these tasks.

7 Discussion

We successfully distilled conceptual information from LLM internals for verbal concepts but not for abstract concepts like "previous" and "next".

Human cognition likely does not process concepts like "next" and "previous" through separate contextual representations. Instead, a shared abstraction—a unified function applied consistently across domains—enables flexible generalization. Investigating whether LLMs exhibit traces of such abstract knowledge, and how to develop it, is critical for achieving human-level artificial reasoning systems.

Limitations

A key limitation is our exclusive focus on linear representations (aligned with the Linear Representation Hypothesis (Elhage et al., 2022; Park et al., 2024)), despite evidence that LLM representations can be nonlinear (Engels et al., 2024). Our LLMs might still encode "Next" and "Previous" nonlinearly but our methods fail to capture it.

Furthermore, Lampinen et al. (2024) notes that assessing model representations using linear methods can prioritize simpler features, even when complex ones are equally well-learned. Even so, the clear differences between verbal and abstract representations, along with the challenges in abstract tasks, support our conclusion that the "previous" and "next" concepts are either not represented or are represented suboptimally.

Finally, our conclusions are restricted to the LLama-3.1 8B and 70B models, leaving generalizability to other architectures untested.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *Preprint*, arXiv:2005.14165.

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. 2022. Toy Models of Superposition. *Preprint*, arXiv:2209.10652.

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

Joshua Engels, Eric J. Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2024. Not All Language Model Features Are Linear. *Preprint*, arXiv:2405.14860.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *Preprint*, arXiv:2404.15255.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-Context Learning Creates Task Vectors. *Preprint*, arXiv:2310.15916.

Felix Hill, Adam Santoro, David G. T. Barrett, Ari S. Morcos, and Timothy Lillicrap. 2019. Learning to Make Analogies by Contrasting Abstract Relational Structure. *Preprint*, arXiv:1902.00120.

Douglas Hofstader. 1979. Gödel, Escher, Bach: An Eternal Golden Braid. *New York: Basic Books*, 11(4):775–792.

Nikolaus Kriegeskorte. 2008. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*.

Andrew Kyle Lampinen, Stephanie C. Y. Chan, and Katherine Hermann. 2024. Learned feature representations are biased by complexity, learning order, position, and more. *Preprint*, arXiv:2405.05847.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Martha Lewis and Melanie Mitchell. 2024. Evaluating the Robustness of Analogical Reasoning in Large Language Models. *Preprint*, arXiv:2411.14215.

Melanie Mitchell. 2020. *Artificial Intelligence: A Guide for Thinking Humans*, first picador paperback edition, 2020 edition. Picador, New York.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The Linear Representation Hypothesis and the Geometry of Large Language Models. *Preprint*, arXiv:2311.03658.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function Vectors in Large Language Models. *Preprint*, arXiv:2310.15213.

Taylor Webb, Keith J. Holyoak, and Hongjing Lu. 2023. Emergent Analogical Reasoning in Large Language Models. *Preprint*, arXiv:2212.09196.