# Benchmarking and Understanding Compositional Relational Reasoning of LLMs

**Ruikang Ni[1*†], Da Xiao[1*‡], Qingye Meng[2], Xiangyu Li[3†], Shihui Zheng[1], Hongliang Liang[1]**

[1]Beijing University of Posts and Telecommunications
[2]ColorfulClouds Technology Co., Ltd.,
[3]ICBC UBS Asset Management
{ni,xiaoda99,shihuizheng,hliang}@bupt.edu.cn
hilbertmeng@gmail.com, li.xiangyu@icbccs.com.cn

## Abstract

Compositional relational reasoning (CRR) is a hallmark of human intelligence, but we lack a clear understanding of whether and how existing transformer large language models (LLMs) can solve CRR tasks. To enable systematic exploration of the CRR capability of LLMs, we first propose a new synthetic benchmark called Generalized Associative Recall (GAR) by integrating and generalizing the essence of several tasks in mechanistic interpretability (MI) study in a unified framework. Evaluation shows that GAR is challenging enough for existing LLMs, revealing their fundamental deficiency in CRR. Meanwhile, it is easy enough for systematic MI study. Then, to understand how LLMs solve GAR tasks, we use attribution patching to discover the core circuits reused by Vicuna-33B across different tasks and a set of vital attention heads. Intervention experiments show that the correct functioning of these heads significantly impacts task performance. Especially, we identify two classes of heads whose activations represent the abstract notion of true and false in GAR tasks respectively. They play a fundamental role in CRR across various models and tasks.

**Dataset and code** — https://github.com/Caiyun-AI/GAR

## Introduction

Compositional relational reasoning (CRR), the ability to reason about multiple types of relations between different entities and combine them to draw conclusions or make predictions, is a hallmark of human intelligence. Transformer large language models (LLMs) have become the de facto backbone for foundation models due to their exceptional performance on various tasks. An important open question is whether and how existing LLMs can solve CRR tasks.

Attempting to answer this question, a number of works study the CRR capabilities of LLMs using *synthetic benchmarks* (Weston et al. 2016; Lake and Baroni 2018; Clark, Tafjord, and Richardson 2020). Compared with real-world benchmarks, synthetic benchmarks offer precise control over the data creation process, helping to understand the strengths and weaknesses of models on targeted tasks. Thus, they are more suitable for *benchmarking* CRR, which occurs relatively rarely in real-world corpora. Besides, we also want to *understand* the underlying mechanism by which the models solve CRR using *mechanistic interpretability* (MI) (Bereska and Gavves 2024) analysis.

One line of work studies compositional or multi-step reasoning (Dziri et al. 2024; Press et al. 2023; Yang et al. 2024; Allen-Zhu and Li 2023; Ye, Li, and Allen-Zhu 2024; Zhang et al. 2022; Sanford, Hsu, and Telgarsky 2024; Brinkmann et al. 2024; Thomm et al. 2024; Zhao and Zhang 2024). They use synthetic tasks to reveal some fundamental limitations of LLMs on such tasks, manifested as either poor generalization when task complexities of training and test set differ (Dziri et al. 2024), or the *compositionality gap* (Press et al. 2023), i.e., how often models correctly answer all subproblems but not generate the overall solution. In these tasks, difficulty is mainly controlled by the number of reasoning steps. When the number becomes large, the over-complex input and usually poor performance of existing LLMs, especially open source ones, make in-depth MI study infeasible. Several studies train specialized models from scratch on the proposed tasks, but the conclusions drawn from analyzing these models may not necessarily apply to existing LLMs. Moreover, the benchmarks used by most of these works have only a single dimension of variety, namely, the number of steps, which captures only an aspect of CRR.

Another line of research uses synthetic tasks for MI study, e.g. associative recall (AR) (Ba et al. 2016; Fu et al. 2023; Olsson et al. 2022), knowledge recall (KR) (Meng et al. 2022; Geva et al. 2023), greater-than (Hanna, Liu, and Variengien 2023) and indirect object identification (IOI) (Wang et al. 2023). Using techniques such as path patching (Wang et al. 2023) and attribution patching (Syed, Rager, and Conmy 2023; Hanna, Pezzelle, and Belinkov 2024), some works discover the circuits - subgraph of the model's computation graph consisting of attention heads and MLPs - responsible for solving the tasks. These works deepen our understanding of the general working mechanism of LLMs. For example, the induction head mechanism found by studying AR tasks underpins the in-context learning capability of Transformer LLMs (Olsson et al. 2022). And the IOI circuit is also generalizable to other tasks (Merullo, Eickhoff,

*These authors contributed equally.
†Contribution during internship at ColorfulClouds Tech.
‡Corresponding author

and Pavlick 2024). While the tasks used in these works are suitable for MI study, they are usually too simple for mainstream LLMs (e.g., a model as small as GPT-2-117M can do AR and IOI tasks perfectly) to reveal any deficiency of LLMs in CRR. They also lack variety; most current MI research is done on a *single* task without any systematic study.

In summary, the lack of a CRR benchmark with both appropriate difficulty and sufficient variety in existing work hinders systematic MI studies on the CRR capabilities of LLMs. In this paper, we make the following contributions:

- We propose a new synthetic benchmark called Generalized Associative Recall (GAR) by integrating and generalizing the essence of several tasks in MI, e.g. associative recall, knowledge recall, indirect object identification (IOI), in a unified framework. GAR consists of a set of automatically generated tasks with varying forms (e.g. affirmative/negative, generation/classification) and difficulties. It is challenging enough to stress the CRR capability of mainstream LLMs, meanwhile simple enough for systematic MI study.
- We evaluate existing LLMs, e.g. open source Llama-2/3 7B-70B, close source GPT 3.5/4, on GAR to show that it is challenging for these LLMs despite appearing simple. Scaling helps but the compositionality gap increases, revealing fundamental deficiency of these LLMs in CRR.
- To understand how LLMs solve GAR tasks, we use attribution patching to discover the core circuits reused by Vicuna-33B across different tasks, and a set of vital attention heads. Intervention experiments show that the correct functioning of these heads significantly impacts task performance. Especially, we identify two classes of heads whose activations represent the abstract notion of true and false in GAR tasks respectively. Experiments show that they play fundamental roles in CRR across various models and tasks. To our knowledge, it is the first time that such heads are identified and studied in real LLMs.

## Generalized Associative Recall Benchmark

### Basic Idea: From AR and KR to GAR

Two well studied synthetic tasks in MI are AR and KR, which inspire GAR. In AR, given a sequence of key-value ($K$-$V$) pairs as context and one of the keys as query $Q$, the model must recall the value associated with the specified key as answer $A$, e.g. "*H 1 C 4 M 7 ... C* $\rightarrow$ *4*".

Compared with AR that requires the model to recall the information in context, KR requires the model to retrieve the factual knowledge stored in its parameters. Given a subject $Q$ and a relation, the model must predict the corresponding attribute $A$, e.g. "*A dog is a kind of* $\rightarrow$ *animal*".

To view these two tasks from a unified perspective, we connect elements $K$, $V$, $Q$, $A$ in them with edges representing two types of relations, as shown in Figure 1 (left): long-range *semantic relations* (solid arrows), e.g. `same`, `kindOf`; local *syntactic relations* (dashed lines), e.g. subject-object, adjacent positions. We observe that for both tasks the two types of relation edges interleave and form a loop[1], which we call a *relational loop*. This is not a coincidence because, from first principles, the existence of the loop ensures predictability.

There are two semantic relations in AR. One (denoted as $r_{lookup}$, $C{\rightarrow}C$ in Figure 1) is used for looking up the correct $K$-$V$ pair in context, the other ($r_{retrieve}$, $4{\rightarrow}4$) is for retrieving the value $V$ and predict answer $A$. In AR, both $r_{lookup}$ and $r_{retrieve}$ are `same` relation. To generalize AR and KR to GAR, we just replace $n_r$ of these two `same` semantic relations (black arrows) in AR with other semantic relations (red arrows) like those in KR (e.g. `kindOf`), where $0 \leq n_r \leq 2$, e.g. "*John has an apple. Mary has a dog ... So Mary has a kind of* $\rightarrow$ *animal*" ($n_r = 1$. $r_{retrieve}$ is replaced). The number of *non*-`same` semantic relations $n_r$ is a key factor for task difficulty in GAR (see Figure 2 (b)). The relational loops remain. We argue that they are the motif of CRR. Besides guiding task design, they are also helpful for understanding the mechanism, as will be shown in Figure 3.

### Workflow for Task and Example Generation

A task of GAR and a basic form example from it with $n_{KV}$ $K$-$V$ pairs are generated in three steps:

- **Step 1:** Select two relational schemas from a predefined set of relational schemas (i.e. sets enriched with some relations between elements), and for each schema select a relation to be used in the next step. This work uses seven relational schemas divided into two types (Table 1): commonsense and factual, ensuring the variety of the tasks;
- **Step 2:** Sample $Q$, $K$ from the domain and codomain of $r_{lookup}$ and similarly sample $V$, $A$ from $r_{retrieve}$ to make the relational loop. Also sample $n_{KV}-1$ elements from the complement set of $\mathrm{domain}(r_{lookup})$ and $\mathrm{domain}(r_{retrieve})$ respectively to form $n_{KV}-1$ distracting $K'$-$V'$ pairs. Then shuffle the $n_{KV}$ $K$-$V$ pairs (not shown in the Figure 1).
- **Step 3:** Convert the data structures obtained in Step 2 into natural language statements by filling templates.

Besides the basic form, we apply two semantic variations and two syntactic variations to increase task variety.

**Semantic variation *negate*** is closely related to IOI (Wang et al. 2023), another well studied task in MI. An example is "*When John and Mary went to store, Mary gave a drink to* $\rightarrow$ *John*". Drawing the relation edges (Figure 1 bottom left), we observe that the essence of IOI is *set complements* (essentially also the logical semantics of *negation*), e.g. {"*John*", "*Mary*"} $-$ {"*Mary*"} = {"*John*"} (assuming the set {"*John*", "*Mary*"} is the universe. $\neg$`same` is used to denote the set complement semantic relation, which means 'not same'/'except'). We incorporate this into GAR to turn an affirmative statement to its negative form by adding a $\neg$`rel` semantic relation (dotted blue arrow) to the relational loop, e.g. "*John has an apple. Mary has a dog ... So Mary doesn't have a kind of* $\rightarrow$ *fruit*". The introduction of negation significantly increases task difficulty.

[1]Interleaved long-range and local relations are also characteristics in other benchmarks for reasoning, e.g. Zhang et al. (2022)

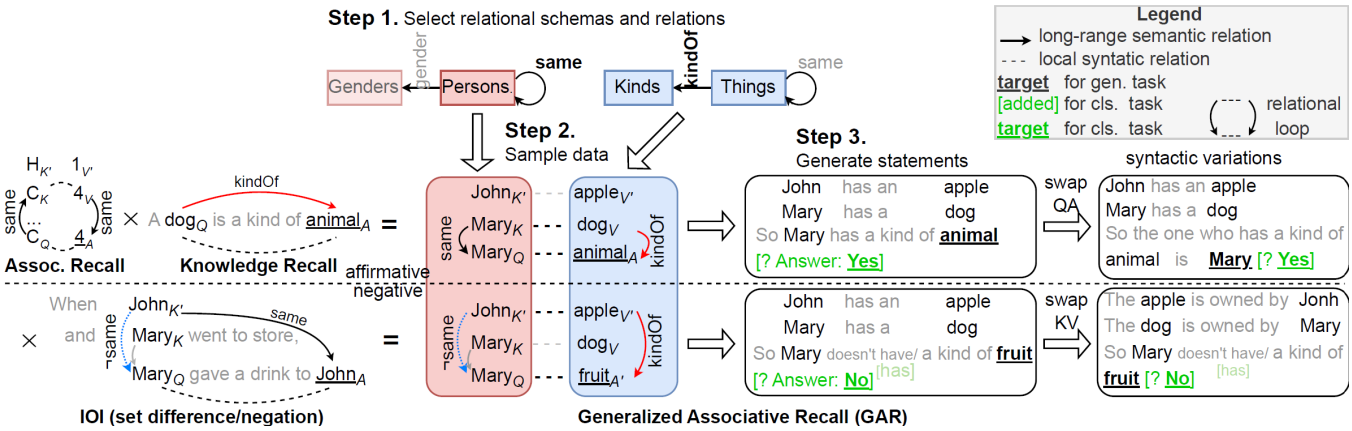



Figure 1: Overall framework of Generalized Associative Recall (GAR).

| Type | Relational schema | $A$ (codomain) | $R_0$ ($\in$) | $B$ (domain) | $R_i$ |
|---|---|---|---|---|---|
| Commonsense | GendersOfPersons | boy, girl | **isA** | John, David, Mary, Ann, ... | **same**, sameGender |
| | KindsOfThings | fruit, animal, ... | **kindOf** | apple, banana, dog, cat, ... | **same**, sameKind |
| | UsagesOfThings | drive, write, ... | **usedFor** | car, truck, pen, chalk, ... | **same**, sameUsage |
| | Adjectives | | | happy, glad, fast, poor, ... | **synonym**, antonym |
| Factual | OccupationOfPersons* | Actor, author, ... | **worksAsA** | Tom Hanks, Stephen King, ... | **same**, sameOccupation |
| | CountriesOfCities | China, France, ... | **inCountryOf** | Beijing, Shanghai, Paris, Lyon, ... | **same**, sameCountry |
| | CountriesOfLandmarks* | China, France, ... | **inCountryOf** | Forbidden City, Louvre Museum, ... | **same**, sameCountry |

Table 1: Relational schemas with form $A \xleftarrow{R_0} B \overset{\{R_i\}}{\circlearrowleft}$, where $\leftarrow$ denotes a one-to-many relation. Relations used in GAR tasks are bolded. Schemas marked with * are from Hernandez et al. (2024). The others are manually constructed by the authors.

**Semantic variation *g2c*** converts the generation task to a classification one. Instead of predicting the last missing $A$, the model must judge the truthfulness of the complete statement and predict *Yes*/*No* for the affirmative/negative form. Examples are given in Figure 1 (green text). The generation task and its corresponding classification task share the same underlying data and relational loop. The former is similar to the LAMBADA benchmark (Paperno et al. 2016), while the latter is similar to NLI tasks, e.g. Bowman et al. (2015), allowing us to study the difference and connection between the mechanisms underlying these two forms of tasks.

**Syntactic variations** *swapQA* and *swapKV* means swapping the order of $Q$ and $A$ or of $K$ and $V$. Examples are shown in Figure 1 (right). These variations change the local syntactic relation, e.g. from subject-object/prev position to object-subject/next position. We use these syntactic variations to investigate if perturbing syntactic relations like this has any impact on model performance.

**Tasks** The two relational schemas with selected semantic relations $r_{lookup}$ and $r_{retrieve}$, number of $K$-$V$ pairs $n_{KV}$ and the applied semantic and syntactic variations jointly define a task. We use the following format for task identifiers:

$r_{lookup}, r_{retrieve} \times n_{KV}$ [*semantic var.*](*syntactic var.*)

e.g. GendersOfPersons/same,KindsOfThings/kindOf×3 [negate](swapKV), where `same` and `kindOf` can be abbreviated as $=$ and $\in$. In this work, $n_{KV}$ defaults to 3. Table 2 shows some tasks with examples. The combination of these different relational schemas and variations results in a total of 384 tasks in GAR. They have varied forms and controllable difficulty levels, while still require some shared basic capabilities of CRR, enabling systematic MI study.

## Evaluating LLMs on GAR

To evaluate if existing LLMs can solve GAR tasks, we test 10 models (Figure 2 (a)). The GAR dataset consists of 192 generation tasks and 192 classification tasks and a total of 4608 examples, with 8/16 examples per generation/classification task. To obtain better performance, all examples are formatted as in-context one-shot learning. Figure 2 (a) shows the average accuracy and predicted probability of the correct answers for both generation and classification tasks. It is evident that generation are harder than classification, and accuracy correlates positively with answer probability, both increasing with model size. Compared with several existing multi-hop reasoning benchmarks (Dziri et al. 2024; Zhang et al. 2022; Sanford, Hsu, and Telgarsky 2024), GAR is two-hop. Though looking simple, due to the introduction of *non*-`same` semantic relations and variations, GAR is still challenging for existing LLMs, even for GPT-4 with an average accuracy of only 71.5%, far below perfect level.

**Task difficulty and compositionality gap** GAR task difficulty can be adjusted by modifying the number of *non*-`same` (other) semantic relations $n_r$ and applying the negate semantic variation. Generation task accuracies with varying difficulty are shown for GPT-4 and Vicuna-33B in Figure 2 (b). The accuracies on simple KR tasks with *other* seman-

| Relational Schemas / Relations | $n_{KV}$ | Semantic Variation | Syntactic Variation | Example |
|---|---|---|---|---|
| KindsOfThings/$\in$ | 0 | | | **Papaya** is a kind of **fruit**. |
| GendersOfPersons/=, CountriesOfCities/$\in$ | 2 | | swapKV | **Madrid** attracts **Michael**. **Bangkok** attracts **John**.<br>So **John** wants to go to a city of **Thailand** |
| OccupationOfPersons/$\in$, UsagesOfThings/$\in$ | 2 | negate | swapKV | The **biro** is **Frida Kahlo**'s. The **telephone** is **Meryl Streep**'s.<br>The **artist** does not have a thing used for **communicating** |
| GendersOfPersons/=, Adjectives/$\sim$ | 3 | g2c | | **Sarah** is **slow**. **Donna** is **rational**. **Steven** is **selfless**.<br>Can we infer that **Steven** is **altruistic**? Answer: Yes |
| GendersOfPersons/=, KindsOfThings/= | 3 | g2c | swapQA | **Tom** has **car**. **Lisa** has **piano**. **John** has **sweater**.<br>The one who has **car** is **Lisa**? Answer: No |

Table 2: Some GAR Tasks with examples.

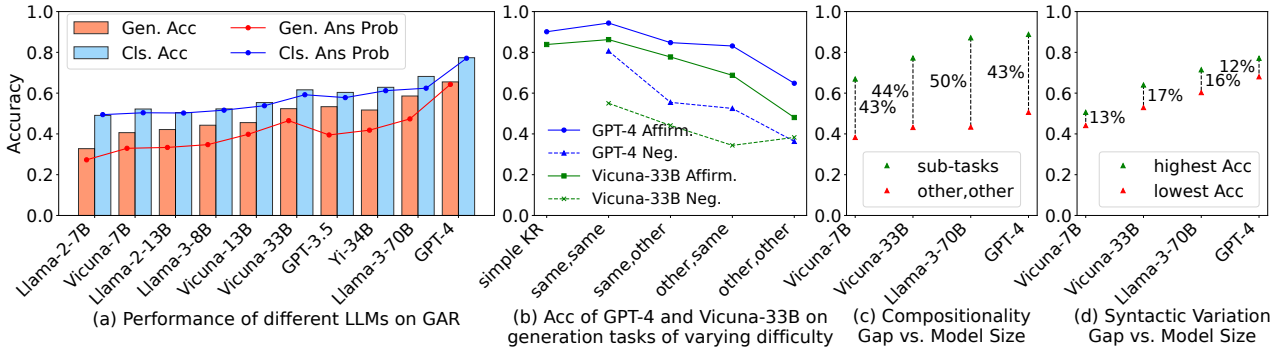



Figure 2: Performance of existing LLMs on GAR.

tic relations (see 1st example in Table 2) are also shown for comparison. Task difficulty increases with $n_r$, and this trend is consistent across models of different sizes. By combining $n_r = 2$ (other, other) and negate (3rd example in Table 2), the accuracies of the models drop below 40%, even for GPT-4. Figure 2 (c) shows the compositionality gaps for the above 4 models on generation tasks. We use simple KR with *other* semantic relations and $n_r$=0 tasks with two `same` semantic relations (same, same), both affirmative and negative, as sub-problems since they contain enough basic knowledge and skill which can be composed to solve the hardest $n_r$=2 tasks (other, other). The compositionality gap is computed as the ratio of the average accuracy of sub-problems to that of $n_r$=2 tasks. The Llama series of models (including Vicuna) show an increasing compositionality gap as the model scales, revealing some fundamental deficiency of these LLMs in CRR. In contrast, syntactic variations have much less impact on performance (Figure 2 (d)), indicating that the models are less sensitive to these perturbations.

## Discovering and Analyzing the Circuits

To understand the general mechanism by which LLMs solve GAR tasks, we do systematic MI study to extract reusable core circuits of Vicuna-33B by comparing individual circuits discovered for solving some tasks. We use step-wise patching similar to Wang et al. (2023), tracing from model outputs back to inputs, but replace path patching with integrated gradients-based attribution patching (Hanna, Pezzelle, and Belinkov 2024) for much faster speed. We use KL divergence as the metric to compute gradient. At each attribution step, we choose among query, key or value attribution manually, roughly following the same logic as Wang et al. (2023). Currently, we do not use any automatic circuit discovery methods because it is hard to make existing methods (Conmy et al. 2023; Hanna, Pezzelle, and Belinkov 2024) work with our model size and circuit complexity. We leave it for future work. We choose the Vicuna-33B model because its performance on GAR tasks is good enough for meaningful MI study while its size guarantees affordable attribution cost. We choose 8 tasks that are representative and that can be solved by Vicuna-33B with high accuracy to ease attribution. Note that Vicuna-33B is larger than most models studied in MI literature with 60 layers and 52 heads. The complete circuits for these tasks are quite complex. We only show the main circuits in Figure 3, omitting some minor branches for clarity.

### Circuits in Classification Tasks

As shown in Figure 3 (a), the first identified contribution to *Yes/No* logits comes from output of an MLP at layer 36, whose input is obtained from the hidden state of token $A$ at layer 15 through an MLP and a $A \rightarrow E$ head. A further step of attribution finds a class of very important heads, namely higher-order relating (*Rel.*$^2$) heads, which includes two True heads 14.18, 14.46 and two False heads 15.51, 14.0, responsible for writing a True/False signal into the residual stream at $A$ according to the relation between $Q$ and $A$.

Query and key attributions of the True and False higher-order relating heads are demonstrated in Figure 3 (b) and 3 (c) separately. For True heads, their query $A$ gathers the

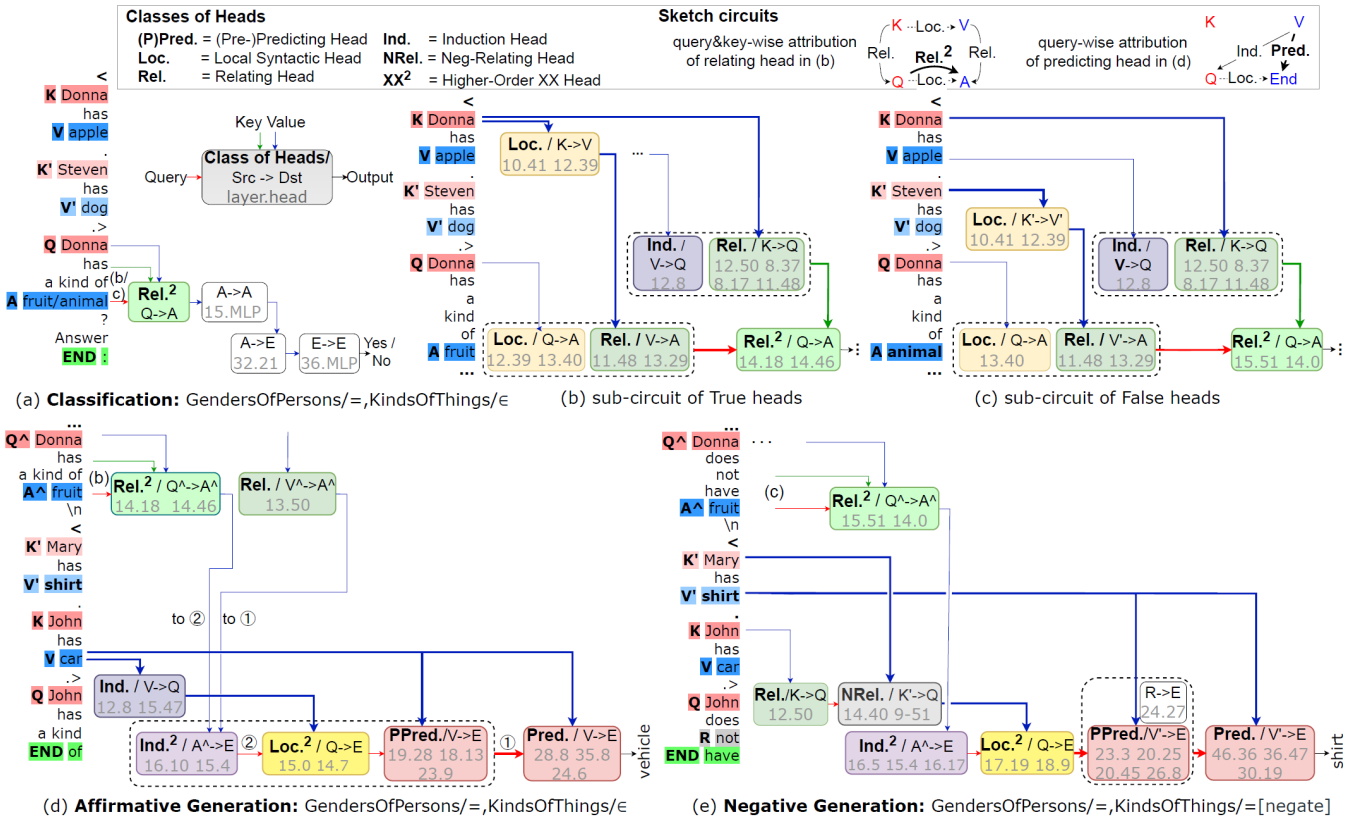



Figure 3: Circuits of Vicuna-33B for solving GAR tasks

information of $Q$ by local heads, and of $K$-$V$ pair by composition of the relating and local heads, while its key $Q$ gathers the information of $K$ and $V$ by the relating and induction heads respectively. Therefore, the information of query side $K$-$V$ pair and the key side $K$-$V$ pair can be compared and matched to affirm that the statement is true. False heads (Figure 3 (c)) share similar circuit as True heads except that 1) the relating heads move the distractive $K'$-$V'$ to $A$; 2) the False heads are activated at $A$ by comparison of query-wise $K'$-$V'$ and key-wise $K$-$V$ pairs, which don't match. In both affirmative and negative circuits, the local $Q \rightarrow A$ heads contribute to the formation of $Q \rightarrow A$ attention pattern of the higher-order relating heads.

## Circuits in Generation Tasks

Circuits in the affirmative and negative generation tasks are depicted in Figure 3 (d) and (e) separately. For the affirmative one, we first identify two classes of heads, predicting and pre-predicting heads. The former is responsible for correctly attending to $V$ ("*car*") and retrieving its `kindOf` attribute ("*vehicle*"), and its query attribution reveals that the formation of its attention pattern relies on the pre-predicting heads. Further query attribution of these (pre-)predicting heads finds the higher-order local ($Loc.^2$, $Q \rightarrow E$) heads, whose value attribution in turn finds the induction heads ($V \rightarrow Q$), through which the information of $V$ flows through $Q$ into $End$, helping form the attention $V \rightarrow E$ of the predicting heads. Query attribution of the $Loc.^2$ heads discovers another class of higher-order Induction ($Ind.^2$) heads, which transmit in-context signals from answer $A^{\wedge}$ in the previous one-shot demonstration to the current token, promoting the activation of the $Loc.^2$ heads ($Q \rightarrow E$) by the higher-order True heads ($Q^{\wedge} \rightarrow A^{\wedge}$) activated on the previous demonstration (path ②), which are exactly the same class of heads we find in the classification circuits (Figure 3 (b)). Similarly, the relating head ($V^{\wedge} \rightarrow A^{\wedge}$) activated on the demonstration promotes the activation of the predicting heads $V \rightarrow E$ (path ①), through the same set of $Ind.^2$ heads. So the higher-order induction heads play a fundamental role in bridging different classes of attention heads across context.

The predicting heads identified in the negative task are different from those in the affirmative task because $r_{retrieve}$ changes from `KindsOfThings` to `same`. With similar attribution we can find different pre-predicting, higher-order local and higher-order relating heads consecutively. Value attribution of the higher-order local heads shows that a relating head ($K \rightarrow Q$) promotes negative-relating heads ($K' \rightarrow Q$) which transmit the information of $K'$ to $Q$ then to $E$, helping the (pre-)predicting heads attend from $E$ to $V'$, which also has the information of $K'$. An $R \rightarrow E$ head attending to "*not*" also has some contribution to the attention formation of the predicting heads. The key difference between the negative and affirmative circuits is that different higher-order relating heads activated on the demonstration activate different higher-order local heads, though via the same higher-order induction heads.

### Core and Sketch Circuits

Putting it all together, we can extract the core circuits reused across various tasks.

- **Truthfulness Sub-Circuit:** The circuit of higher-order relating heads (Figure 3 (b) and (c)) shared between classification and generation tasks detects the higher-order relation between $A$ and $Q$ to judge truthfulness.
- **ICL Sub-Circuit:** Across various generation tasks, different (higher-order) relating heads activate different higher-order local heads or predicting heads through the same higher-order induction heads, which plays a fundamental role in in-context learning.
- **Overall Circuit:** The truthfulness sub-circuit can be combined either with downstream MLPs to complete classification tasks (Figure 3 (a)), or with downstream attention heads via the ICL sub-circuit to solve generation tasks (Figure 3 (d), (e)).

To understand these circuits from the *relational loop* perspective, the first sketch in Figure 3 (legend) shows query-wise (from $A$) and key-wise (from $Q$) attribution of the higher-order relating heads in Figure 3 (b), where two links between $Q$ and $A$ appear: one is formed directly by the local heads; the other is formed indirectly by composition of query-wise $K \rightarrow V \rightarrow A$ and key-wise $K \rightarrow Q$. The two links together form a relational loop, which are *detected* by the higher-order True heads. The second sketch shows a high-level sketch of the circuit in Figure 3 (d). The critical attention pattern $V \rightarrow E$ of the predicting heads is formed by composition of $V \rightarrow Q$ and $Q \rightarrow E$, which *completes* the relational loop required for solving the generation task.

## Validating Attention Heads

In this section, we use intervention and other methods to validate the vital roles of several types of heads identified in the previous section. We first explain a few concepts:

- **Head Activation**: If at any attending position (a row in the attention weight matrix) a head assigns the largest attention weight to any token other than the first token $<s>$, this head is considered to be *activated*.[2] The largest such weight across the whole attention matrix is defined as its *activation value*.
- **True/False Heads**: Higher-order relating heads which activate on true/false statements for truthfulness judgement (Figure 3 (b) / (c)).
- **Strong Intervention**: This intervention forces the attention weights of a head to fully comply with the expected attention pattern, e.g. for attention pattern $V \rightarrow E$, the attention weight of token $End$ attending to token $V$ is set to 1 while the weights to the other tokens are set to 0.
- **Weak Intervention**: This intervention replaces the attention weights of a head at the attending position (e.g. $End$) with weights found from all other heads in the model that complies best with the attention pattern (e.g. $V \rightarrow E$). Unlike strong intervention which directly imposes an attention pattern (essentially the same as the intervention method used in Merullo, Eickhoff, and Pavlick (2024)), weak intervention utilizes the attention weights already formed by the model itself, though from other heads, thus avoiding the introduction of additional information from outside the model.

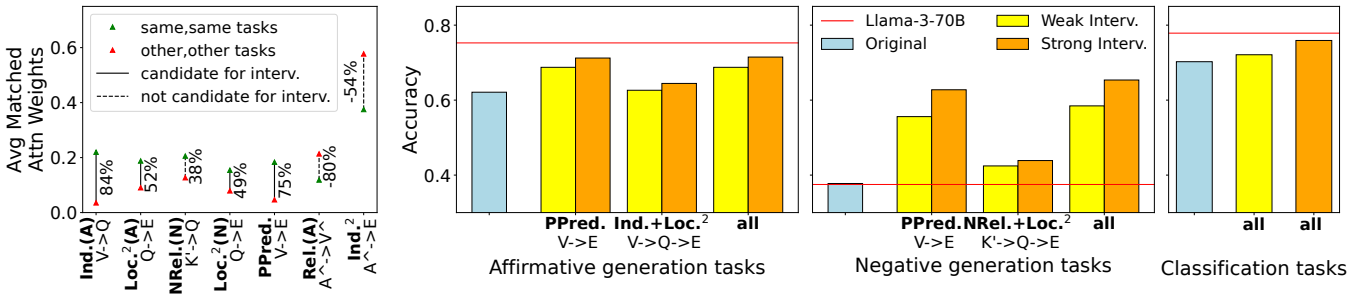
[2]We observed that for Llama/Vicuna models the first token plays the role of "sink token" for null attention (Xiao et al. 2024).

### Validating Non-True/False Heads in Vicuna-33B

We apply weak and strong interventions to several classes of attention heads in Figure 3 to validate their importance. As a motivation for intervention, we first inspect their average attention weights in generation tasks that match the desired attention pattern, e.g. the average weight of $End$ attending to $V$ for pattern $V \rightarrow E$, which measures how well the heads function as expected. As shown in Figure 4 (a), similar to the compositionality gap, the matched attention weights of some classes of heads also exhibit a clear gap between the easiest same,same tasks and the hardest other,other tasks (recall discussion on task difficulty and compositionality gap in Figure 2 (c) for definition of these tasks), partially explaining the performance drop on harder tasks. In contrast, higher-order induction heads are more robust, maintaining high matched attention weights across tasks.

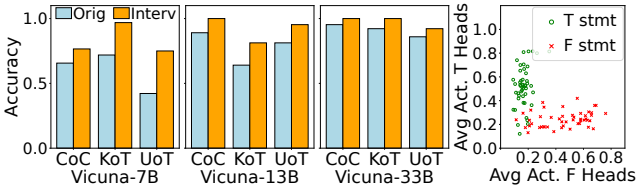
For generation tasks, we intervene on the class of heads with positive matched attention weight gaps in Figure 4 (a). We jointly intervene on higher-order local heads with induction heads (Ind.+Loc.$^2$, for affirmative tasks) or with negative-relating heads (NRel.+Loc.$^2$, for negative tasks), because they V-compose to form critical paths in the circuits (Figure 3 (d,e)). For classification tasks, we jointly intervene on all relating heads and induction heads, because they work in parallel to form the attention of True/False heads (Figure 3 (b,c)). We don't intervene on 1) predicting heads, which vary between different $r_{retrieve}$ for generation tasks and are thus not universal, and 2) relating heads in affirmative generation tasks and higher-order induction heads, which exhibit negative matched attention weight gaps.

Figure 4 (b) shows that strong intervention jointly on all heads increase the accuracy of Vicuna-33B significantly, approaching (for affirmative generation and classification tasks) or even surpassing (for negative generation tasks) the performance of Llama-3-70B. For generation tasks, intervention on pre-predicting heads is most effective, which is intuitive because they are closest to and thus have the most direct impact on the final output of the model. Intervening on the other heads (Ind.+Loc.$^2$ in affirmative tasks and NRel+Loc.$^2$ in negative tasks) has smaller effect, perhaps because they are too far away from the final output to have strong impact. To conclude, the effectiveness of strong intervention shows that the correct functioning of these heads, especially correct formation of the required attention patterns, significantly impact task performance. Weak intervention is also effective, albeit weaker, indicating that the model has the intrinsic ability to attend correctly, but with other heads that cannot compose the functional circuits, showing potential for improvement.




(a) Matched attn weight gaps (b) Accuracies of Vicuna-33B by intervening different classes of vital heads

Figure 4: Analysis and intervention results of some vital heads in Vicuna-33B. In Figure (a), (A) denotes affirmative generation tasks and (N) denotes negative generation tasks.




(a) Acc. of Vicuna models by intervening True/False heads (b) Act. of True/False heads distinguish True/False stmt.

Figure 5: Analysis and intervention of True/False heads. CoC = CountriesOfCities, KoT = KindsOfThings, UoT = UsagesOfThings

## Validating True/False Heads across Models

After validating the upstream heads of the True/False heads in Vicuna-33B (Figure 4 (b) rightmost), we further validate the effectiveness and universality of the True/False heads themselves by intervention on different-sized vicuna models. We conduct experiments on three representative tasks: GendersOfPersons/=, $r_{retrieve}/\in$, where $r_{retrieve} \in$ {CountriesOfCities (CoC), KindsOfThings (KoT), UsagesOfThings (UoT)}. Using the attribution method described previously, we identify the True/False heads in different-sized Vicuna models for intervention. Specifically, when the answer is *Yes*/*No*, we apply intervention on the True/False heads while knocking out the False/True heads. As shown in Figure 5 (a), after intervention, the average accuracy of Vicuna-7B/13B/33B is increased by 17%/14%/6%, indicating that the True/False heads are universal and exhibit consistent effects across different model sizes.

To understand why Vicuna-33B already has high classification accuracy before intervention, we randomly sample 96 examples from the classification tasks and plot each example as a dot using the average activation values of the True heads and False heads as coordinates in Figure 5 (b). It can be seen that the activation values can effectively distinguish true and false statements, indicating that these True/False heads definitely represent the abstract notion of true and false in these tasks. Combining Figure 5 (a) and (b), it is evident that the activation status of True/False heads encodes the truthfulness of statements and that the models indeed use them to judge truthfulness in GAR tasks.

## The Efficacy of True/False Heads in Other Datasets

To investigate if the True/False heads also play an important role on other datasets besides GAR, we test them on two datasets that require to judge if a statement is true (entailment) or false (contradiction): Stanford Natural Language Inference (SNLI, Bowman et al. (2015) and Geometry of Truth (GoT, Marks and Tegmark (2023). We forward the examples through Vicuna-33B and extract the activation values of four True/False heads (14.18, 14.46, 15.51, 14.0) as features to train simple MLP classifiers to predict true or false.

Table 3 shows the results along with one-shot accuracies of several other different-sized models on these two datasets for comparison. The MLP classifiers' accuracies approach (SNLI) or surpass (GoT) Vicuna-7B, indicating that the activations of these heads encode information for truthfulness classification. The activation patterns of these True/False heads identified in GAR are robust across other datasets.

| Model | SNLI Acc(%) | GoT Acc(%) |
|---|---|---|
| Random Guess | 50 | 50 |
| GPT-2-Medium (345M) | 51.43 | 51.04 |
| GPT-2-XL (1.5B) | 53.90 | 50.35 |
| **True/False Head Act + MLP** | **87.07±0.2** | **85.63±1.8** |
| Vicuna-7B | 91.00 | 84.73 |
| Vicuna-33B | 96.80 | 89.69 |

Table 3: Accuracy of different models and MLP classifiers with activation values of True/False heads of Vicuna-33B as features on SNLI and on 4 subsets of GoT.

## Conclusion

We propose the Generalized Associative Recall (GAR) benchmark that is challenging enough to stress the CRR capability of mainstream LLMs, meanwhile simple enough for systematic MI study. We evaluate existing LLMs on GAR to show that the compositionality gap increases despite scaling, revealing fundamental deficiency of these LLMs in CRR. We discover the core circuits reused by Vicuna-33B across different GAR tasks and a set of attention heads important for task performance, especially the True/False heads, which can represent true/false statements in GAR tasks and play fundamental roles in CRR across various models and tasks.

## Acknowledgements

The work was partially supported by National Natural Science Foundation of China (NSFC) (Grant No. 62425105, 62350001, 62206019), Fundamental Research Funds for the Central Universities (Grant No. 530424001) and Taiyuan City "Double hundred Research action" 2024TYJB0127.

## References

Allen-Zhu, Z.; and Li, Y. 2023. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*.

Ba, J.; Hinton, G. E.; Mnih, V.; Leibo, J. Z.; and Ionescu, C. 2016. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29.

Bereska, L.; and Gavves, E. 2024. Mechanistic Interpretability for AI Safety–A Review. In *Transactions on Machine Learning Research (TMLR)*.

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Brinkmann, J.; Sheshadri, A.; Levoso, V.; Swoboda, P.; and Bartelt, C. 2024. A mechanistic analysis of a transformer trained on a symbolic multi-step reasoning task. In *Findings of the Association for Computational Linguistics ACL 2024*.

Clark, P.; Tafjord, O.; and Richardson, K. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*.

Conmy, A.; Mavor-Parker, A.; Lynch, A.; Heimersheim, S.; and Garriga-Alonso, A. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 16318–16352.

Dziri, N.; Lu, X.; Sclar, M.; Li, X. L.; Jiang, L.; Lin, B. Y.; Welleck, S.; West, P.; Bhagavatula, C.; Le Bras, R.; et al. 2024. Faith and fate: Limits of transformers on compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Fu, D. Y.; Dao, T.; Saab, K. K.; Thomas, A. W.; Rudra, A.; and Ré, C. 2023. Hungry hungry hippos: Towards language modeling with state space models. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

Geva, M.; Bastings, J.; Filippova, K.; and Globerson, A. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hanna, M.; Liu, O.; and Variengien, A. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

Hanna, M.; Pezzelle, S.; and Belinkov, Y. 2024. Have Faith in Faithfulness: Going Beyond Circuit Overlap When Finding Model Mechanisms. In *Conference on Language Modeling (COLM)*.

Hernandez, E.; Sharma, A. S.; Haklay, T.; Meng, K.; Wattenberg, M.; Andreas, J.; Belinkov, Y.; and Bau, D. 2024. Linearity of relation decoding in transformer language models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Lake, B.; and Baroni, M. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2873–2882. PMLR.

Marks, S.; and Tegmark, M. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 17359–17372.

Merullo, J.; Eickhoff, C.; and Pavlick, E. 2024. Circuit component reuse across tasks in transformer language models. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Olsson, C.; Elhage, N.; Nanda, N.; Joseph, N.; DasSarma, N.; Henighan, T.; Mann, B.; Askell, A.; Bai, Y.; Chen, A.; et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Paperno, D.; Kruszewski, G.; Lazaridou, A.; Pham, Q. N.; Bernardi, R.; Pezzelle, S.; Baroni, M.; Boleda, G.; and Fernández, R. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Sanford, C.; Hsu, D.; and Telgarsky, M. 2024. Transformers, parallel computation, and logarithmic depth. *arXiv preprint arXiv:2402.09268*.

Syed, A.; Rager, C.; and Conmy, A. 2023. Attribution patching outperforms automated circuit discovery. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Thomm, J.; Terzic, A.; Karunaratne, G.; Camposampiero, G.; Schölkopf, B.; and Rahimi, A. 2024. Limits of Transformer Language Models on Algorithmic Learning. *arXiv preprint arXiv:2402.05785*.

Wang, K.; Variengien, A.; Conmy, A.; Shlegeris, B.; and Steinhardt, J. 2023. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*.

Weston, J.; Bordes, A.; Chopra, S.; Rush, A. M.; Van Merriënboer, B.; Joulin, A.; and Mikolov, T. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; and Lewis, M. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Yang, S.; Gribovskaya, E.; Kassner, N.; Geva, M.; and Riedel, S. 2024. Do Large Language Models Latently Perform Multi-Hop Reasoning? *arXiv preprint arXiv:2402.16837*.

Ye, T.; Li, Y.; and Allen-Zhu, Z. 2024. Physics of language models: Part 3.2, grade-school math and the hidden reasoning process. *arXiv preprint arXiv:2407.20311*.

Zhang, Y.; Backurs, A.; Bubeck, S.; Eldan, R.; Gunasekar, S.; and Wagner, T. 2022. Unveiling transformers with lego: a synthetic reasoning task. *arXiv preprint arXiv:2206.04301*.

Zhao, J.; and Zhang, X. 2024. Exploring the limitations of large language models in compositional relation reasoning. In *Conference on Language Modeling (COLM)*.