Sources of Hallucination by Large Language Models on Inference Tasks

Nick McKenna[†]* **Tianyi Li**[†]*
Liang Cheng[†] **Mohammad Javad Hosseini**[‡] **Mark Johnson**[§] **Mark Steedman**[†]
[†]University of Edinburgh [‡]Google Research [§]Macquarie University
{nick.mckenna, tianyi.li}@ed.ac.uk

Abstract

Large Language Models (LLMs) are claimed to be capable of Natural Language Inference (NLI), necessary for applied tasks like question answering and summarization, yet this capability is under-explored. We present a series of behavioral studies on several LLM families (LLaMA, GPT-3.5, and PaLM) which probe their behavior using controlled experiments. We establish two factors which predict much of their performance, and propose that these are major sources of hallucination in generative LLM. First, the most influential factor is memorization of the training data. We show that models falsely label NLI test samples as entailing when the hypothesis is attested in the training text, regardless of the premise. We further show that named entity IDs are used as "indices" to access the memorized data. Second, we show that LLMs exploit a further corpus-based heuristic using the relative frequencies of words. We show that LLMs score significantly worse on NLI test samples which do not conform to these factors than those which do; we also discuss a tension between the two factors, and a performance trade-off.[1]

1 Introduction

Large Language Models (LLMs) such as LLaMA, GPT-3/4, PaLM, etc. (Touvron et al., 2023; Brown et al., 2020; Chowdhery et al., 2022), have been trusted by many to perform language understanding in downstream tasks such as summarization, question answering, and fact verification, among others (Zhang et al., 2023). However, due to the large-scale nature of LLM training on vast, often proprietary data, and the inherent opacity of LLM parameters, it is difficult to explain their behavior when answering user queries and the corresponding risks in terms of bias and robustness. In particular, one LLM behavior poses a significant challenge: "hallucination," the phenomenon in which LLMs provide information which is incorrect or inappropriate, presented in a factual manner.

This paper investigates how LLMs perform on natural language inference tasks, sometimes called *textual entailment*, a basic capability forming part of language understanding, which is used in real tasks. We look at *directional entailments*, which hold in one direction, but not both, for example, DEFEAT *entails* PLAY but PLAY *does not entail* DEFEAT. Inferring directional entailment is more difficult than that of paraphrase, which is symmetric, so it more deeply probes understanding.

Our approach is a behavioral study of prompted LLM decision-making. We alter existing directional inference datasets in targeted ways while measuring how predictions change, across several major LLM families (LLaMA, GPT-3.5, and PaLM). We demonstrate two sources of LLM performance on the directional NLI task, which also explain false positive hallucination: (1) LLM bias toward affirming sample hypotheses that are attested in the training text, including reliance on named entity identifiers; and (2) a corpus-term-frequency heuristic, biased toward test samples with premises less frequent than hypotheses.

We establish that these originate from the LLM pretraining objective, in which statistical modeling of the natural distribution of human-generated text leads to (at the level of sentences) memorizing individual statements, and (at the level of corpora) learning typical patterns of usage. Though superficially impressive in performance, our experiments show that even powerful LLMs still use unsatisfactory tools instead of human-like reasoning.

We present three contributions, the demonstrations of both factors across several experiments, and an impact analysis:

(1) In a prompting scenario, LLMs respond to NLI test samples according to a *veracity* bias, af-

*Co-first authors with equal contribution.

[1]Code and LLM outputs (from LLaMA and GPT-3.5) are available at https://github.com/Teddy-Li/LLM-NLI-Analysis.

firming hypotheses more readily if seen in the pre-training text. When a hypothesis proposition is attested in training according to the model itself, LLaMA-65B, GPT-3.5, and PaLM-540B are respectively 1.9, 2.2, and 2.0 times more likely to wrongly predict a false positive inference, compared to when not attested. Further, LLMs recall this memorized information using named entities as identifying "indices," even though these are irrelevant to the logic of the predicate inference task.

(2) When relevant memorized text is not available, LLMs back off to a simple corpus-based heuristic using term-frequency to make judgements. LLaMA-65B, GPT-3.5, and PaLM-540B are 1.5, 1.7 and 2.0 times more likely to wrongly predict a false positive if the hypothesis has higher relative frequency than the premise, than if it does not.

(3) For the NLI subsets consistent with these factors, LLMs appear to be excellent classifiers; for NLI subsets adversarial to them, LLM performance degrades severely. We show that when labels go against the *veracity prior*, LLMs degrade into poor or even near-random classifiers; for the *relative frequency heuristic*, we also show a substantial performance decrease with all the LLMs consistently.

2 Related Work

Prior work has addressed the robustness issues of NLI datasets. Poliak et al. (2018) found a range of NLI datasets to be subject to a hypothesis-only bias, whose labels are predictable by supervised models trained on only the hypothesis. In this paper, we use a similar hypothesis-only test with LLMs, but with few-shot examples, to probe model memory without training.

Li et al. (2022) show that smaller Language Models such as RoBERTa (355M parameters) (Liu et al., 2019), under the pre-train-fine-tune paradigm, are reliant on dataset artifacts when performing directional predicate inference. In this paper, we study the behavior of a class of much larger Language Models, which have previously demonstrated more robust performance across NLP tasks.

Recent work has also explored LLM memorization and generalization. Carlini et al. (2023) establish that LLMs are able to memorize more data than small LMs, whereas Tirumala et al. (2022) further hypothesize that LLMs pay special attention early in training to numbers and nouns, which may act as unique identifiers for individual sentences in training. We further show that entity recall is used in language inference, and frequency in training data affects generalization in various ways (§6, §7).

Webson and Pavlick (2022) show that LLMs perform surprisingly well in NLP tasks even conditioned on pathologically unhelpful prompts, calling into question whether LLMs understand prompts the same way as humans. In this vein, we show that tasks formatted for language inference may be answered like memory recall tasks instead, which may happen to align with correct labels.

Recently, Bubeck et al. (2023) advocated that GPT-4 has a deep and flexible understanding of language "far beyond memorization". Although we do not disprove the existence of such abilities, we show that GPT-4, like the other LLMs, is also subject to these hallucinations (see Appendix D).

Corpus frequency statistics have long been studied: it is well-known that nouns follow a trend of becoming more specific as corpus-frequency decreases (Caraballo and Charniak, 1999). McKenna and Steedman (2022) also argued that more specific predicates tend to be less corpus-frequent than more general predicates. Since entailments may carry from a specific predicate to a more general one, e.g. SPRINT *entails* RUN, relative corpus frequency can be indicative of entailment, though it has no direct relationship to meaning.

3 Experimental Design

We design behavioral experiments on LLMs by modifying an original NLI dataset in various conditions, and observing the change in model response. By controlling for each tested factor, we demonstrate significant behavior change across three major LLM families due to memory effects in §5 and §6, and a corpus frequency heuristic in §7. Finally, we show the impact on task performance in §8.

We now define the two priors explored in this paper, as well as the dataset and test conditions in which we evaluate each LLM.

3.1 Veracity and Relative Frequency Priors

The Veracity Prior indicates when a propositional statement is likely to be attested in some way by LLM training data. We measure the attestation of a statement using the LLMs' own response when prompted to predict the veracity of a hypothesis proposition: "[hypothesis]. is this true or false?".[2]

[2]One alternative is to use the LLM's perplexity for a given proposition; however, since perplexity is not available with GPT-3 series models, for even evaluation across models we use the veracity predictions instead.

Veracity predictions are denoted with V.

As discussed in §2, we draw inspiration from the *hypothesis-only baseline* (Poliak et al., 2018), but we use the test to probe model memory without training. We describe prompt generation in more detail in §4.2, and appendix Table 8 shows an example.

The Relative Frequency Heuristic is a simple corpus-based heuristic, assigning a label of `Entail` if the premise is less corpus-frequent than the hypothesis. This heuristic is reflected in the natural distribution of predicates in text due to the anti-correlation of specificity and frequency (more specific terms tend to be less corpus-frequent), and the fact that specific terms may entail more general terms (McKenna and Steedman, 2022). However, this effect has no direct relationship with meaning, thus is risky to trust.

This heuristic would ideally be measured according to an LLM's pre-train corpus, however, these are impractically large and/or proprietary. Instead, we use Google N-grams[3] as a proxy of the natural distribution of text, and thus the distributions of these corpora. We take average frequencies between the years 1950-2019, and compare between the premise P and the hypothesis H.

To account for noise in corpus distributions, we lemmatize each predicate, discard surrounding context words, and require a wide margin of difference between P and H frequencies.

Predictions from this prior are denoted with F: if H is at least 5x more frequent than P, we label it $F = win$; if H is at least 5x less frequent than P, we label it $F = lose$; anything in-between is a draw and left out of F-analyses.

3.2 Dataset

Levy/Holt For our experiments, we use the Levy/Holt dataset, containing premise-hypothesis pairs with a task formatted: "Given [premise P], is it true that [hypothesis H]?". Each P- and H-statement has the property of containing one predicate with two entity arguments, (where the same entities appear in both P and H) as shown in Table 1, so entailment is decidable on the basis of the predicate and its attributes. We study the challenging subset of 1,784 directional questions, where entailments hold in one direction but *not* both.

We aim to test LLMs on their capability to reason purely about the semantics of natural language predicates, not any task relating to knowledge of the world, so we explicitly avoid datasets such as MMLU (Hendrycks et al., 2021), Natural Questions (Kwiatkowski et al., 2019), OpenBookQA (Mihaylov et al., 2018) etc.

3.3 Dataset Transformations

The standard inference task I is presented by the Levy/Holt NLI dataset, in which the answer is determinable using only general language inference of predicates and their attributes. As a task over sentences, each sample sentence also contains entity arguments.

We define three dataset transformations designed to remove aspects of information from original samples. We study the change in model behavior as targeted information is removed. We define three new transformed tasks: randomized premise predicate I_{RP}, type-arguments I_{TA}, and type-constrained randomized arguments I_{RA}.

Each transformation involves first identifying the types of entities in statements in order to constrain replacements in a natural way which is purposefully meaning-preserving or -destructive. To type the entities in each statement, we derive their FIGER type (Ling and Weld, 2012) such as "person," "location," "organization," etc. This is done using an entity linker (Nguyen et al., 2014) which identifies an entity's Freebase ID (Bollacker et al., 2008), from which we obtain a type $t \in \mathcal{T}$ among the 48 FIGER types + 1 default type "thing" used in failure cases.

The random premise task I_{RP} replaces the original premise predicate with a random predicate, while maintaining the same entity arguments. This test is designed to break the link between the premise and hypothesis, since a randomly sampled premise predicate is very unlikely to entail the hypothesis in all cases. We aim to test model sensitivity when entities remain the same, but the predicates are no longer semantically related.

To maintain naturalness and grammaticality, we insert a new predicate that satisfies the same entity type-constraints as the old one: the replacement should have argument slots of the same types as the original premise. For example, "[medicine] is indicated for patients with [disease]" is swapped for "[medicine] does not cure [disease]". We map the entities to their respective slots between the original premise and the replacement. I_{RP} is a good test of generalizing language understanding

[3] https://books.google.com/ngrams

since new strings are created which we assume the model has not seen before in training. An example is shown in Table 1.

To create the I_{RP} task, we source candidate premises from the full Levy/Holt development set of 5,486 questions, and sample uniform randomly from the predicates satisfying the target type-constraints. In this task, entailment labels are all assumed to be `No-Entail`, since a randomly sampled premise is unlikely to entail the hypothesis.

The type-argument task I_{TA} replaces the original arguments with unique, typed identifiers, e.g. "location X" or "food Y". In using basic FIGER types to mask the identities of arguments, this test is designed to remove extraneous information while maintaining the same entailment label, as a baseline control setting. We append unique identifiers "X" and "Y" to allow tracking of entity slots across the premise and the hypothesis, in case statements involve two arguments of the same type.

The random argument task I_{RA} builds off of I_{TA} by replacing original entities with other real, random entities of the same type. Like I_{TA}, this test is designed to modify statements without changing entailment truth values, and tests model sensitivity to novel extraneous information. To this end, we replace the arguments only with entities of the same FIGER types. We use the same mapping across paired samples which test both directions of entailment: forwards ($a \vDash b$) and reverse ($b \nvDash a$). Examples for all argument transformations are shown in Table 3.

I_{RA} is also designed to create novel strings that are unlikely to be in pre-training data. However, the truth value of dataset entailments is not changed, since sample labels are determinable using only the predicate. The entity type constraints additionally ensure polysemous predicates maintain the same sense. For example, a different sense of *run* can be read from "[person] runs [organization]" and "[person] runs [software]"; but among different entities of the same types, predicate senses are consistent, so the exact entity IDs do not affect general inference about predicates.

We source new entities from NewsCrawl, a decade-long span of multi-source news text, in which entities are linked and typed as above. This corpus is used and described in other work (McKenna et al., 2021; Li et al., 2022). We draw new entities uniform randomly from the 5% least frequently mentioned entities in NewsCrawl ($I_{RA} \downarrow$), and the 5% most frequent ($I_{RA} \uparrow$). We swap the arguments for sampled entities while preserving the rest of each statement.

4 Querying Models with Prompts

We describe our methodology for model selection, prompt development, and model scoring.

4.1 Models

GPT-3 Series Though closed to deep scientific review, these are a widely-used comparison due to their performance, and have been reasonably well-studied. We take text-davinci-003 (GPT-3.5) as the primary subject of evaluation (Brown et al., 2020), as it is the largest and best-aligned version.

LLaMA A recent LLM model family which rivals or surpasses GPT-3 performance while being open to scientific study. LLaMA provides a range of model sizes; we test with the largest 65B model. LLaMA is not fine-tuned; while there have been efforts to fine-tune them for alignment with humans (Taori et al., 2023; Chiang et al., 2023), we did not find these to be significantly different from the original on our task, so we leave them out.

PaLM One of the largest available LLM families, we test with the largest 540 billion parameter model, which often claims state-of-the-art on evaluation datasets (Chowdhery et al., 2022). As it is only pretrained, this model serves as a further comparison point to LLaMA.

Later GPT models such as text-davinci-003, used in our experiments, have trained in several phases including pretraining, instruction-tuning, and human alignment via Reinforcement Learning through Human Feedback (RLHF), while base LLaMA and PaLM models have only undergone pretraining, so their contrast indicates what stage of training is responsible for observed phenomena.

Finally, we omit experimenting on open models superseded in performance by LLaMA, such as OPT, GPT-J, etc., and also models which are closed to scientific reviews, such as GPT-4, Bard, etc. [4]

4.2 Prompt Design and Evaluation

Formatting We feed each test sample into the model by insertion of the premise and hypothesis

[4] In appendix D, we also report the impact of the two factors on GPT-4, the LLM receiving the most spotlight; we show that GPT-4 shares the same fragilities as other LLMs.

into a prompt template, which is used to query the model in natural language. Following this, we append a three-way answer choice: A) Entailment, B) Neutral, C) Contradiction, following the typical format in NLI (Bowman et al., 2015).

Tuning We tune our prompt templates on the Levy/Holt dev set. We use a set of 4 most promising prompt templates including the best template from Schmitt and Schütze (2021), also used in other NLI work[5] (Webson and Pavlick, 2022). We choose the best-performing prompt template on the dev set to deploy on the test set.

Ideally, an LLM with advanced language understanding capability could perform inference in zero-shot without any annotated examples, which would raise confidence that this faculty is readily available in downstream tasks such as summarization or QA. To this end, we test GPT-3.5 and LLaMA in zero-shot on the datasets, but they exhibit severely degraded performance, even near random chance.

We turn to few-shot, and hand-annotate a minimal 4 examples in the style of the template, with added explanations about why the given answer is correct for each example. These examples are prepended before the query (see Appendix A for an example prompt). While the definition of "few-shot" varies between works, we seek to use the minimum number of examples necessary to evoke a positive response on the dataset for each model; our goal is to study model behavior as conditions change, not to maximize the score on any particular dataset. We observe that given 4 high-quality annotated examples, each model is able to score very well on the dev portion, across most templates.

Scoring We convert choice A into `Entail` and collapse both B and C choices into `No-Entail` to align with Levy/Holt annotation. For experiments in §5, §6, and §7, we score the model solely based on its textual response. Inspection of dev responses shows the models are compatible with the QA format and respond using A/B/C 100% of the time.

For experiments in §8 which measures practical model performance across confidence thresholds, we convert the letter choice to an "entailment score" with the mapping:

$$S_{\text{ent}} = 0.5 + 0.5 * \mathbb{I}[\text{tok} = \mathbf{A}] * S_{\text{tok}} - 0.5 * \mathbb{I}[\text{tok} \in \{\mathbf{B}, \mathbf{C}\}] * S_{\text{tok}}$$

[5]See Appendix A for the full list of prompt templates.

Where $\mathbb{I}$ is the indicator function, and S_{ent} estimates the probability of positive classification (`Entail`) from a textual output ($0 \leq S_{\text{ent}} \leq 1$) with token probability S_{tok} using a linear transformation, which preserves the ordering of model confidences; this is sufficient for calculating a precision-recall curve, so full probability distributions over all answer tokens (which are not always provided by e.g. GPT) are not necessary.

5 Experiment 1: Dominance of Veracity Prior

We first assess LLMs' reliance on memorization of training text by conditioning each model's entailment task predictions on its own predictions for hypothesis veracity, which measures if a hypothesis is attested in training data. We examine two scenarios: the standard inference task (I), and the random premise task (I_{RP}). We compare model probabilities of predicting `Entail` conditioned on whether the hypothesis veracity is predicted as `True` or not, in order to evaluate the model outputs' dependence on the *veracity prior*.

We further control for the possibility that original Levy/Holt entailments may coincidentally refer to attested facts, which could lead to spurious correlation between inference and veracity scores without demonstrating clearly use of memory versus entailment capability. We apply this control by comparing to the random premise task I_{RP}, which converts entailments into non-entailments without altering the hypothesis. An ideal model capable of language understanding using the information provided in context should detect that in the I_{RP} task it is no longer possible to infer the hypothesis based on the premise (even if the hypothesis is itself attested in training), and never predict `Entail`. Thus, in the I_{RP} task, all `Entail` predictions are assumed to be false positive hallucinations.

5.1 Results

With I, I_{RP} and V predictions acquired as described in §3.1, we present the results in Table 2. It's clear that a model's memory about the hypothesis plays a part in its predictions when conditioned on a premise, either related or random.

For I, we observe significantly increased probability of predicting `Entail` when the hypothesis is previously judged veracious.

In the random premise task I_{RP}, this trend continues. LLaMA, GPT-3.5, and PaLM, respectively,

Task	Dev Sample	
I	(True)	George Bush was the Governor of Texas ⇒ George Bush is a politician from Texas
I_{RP}	(False)	George Bush resided in Texas ⇒ George Bush is a politician from Texas

Table 1: From the original dataset task (I) we derive the Random Premise task (I_{RP}), where the premise predicate has been randomized while respecting type-constraints. A random premise predicate is highly unlikely to entail the hypothesis, so all labels become `False`. Sample format: (`Label`) [premise] ⇒ [hypothesis].

Predictor	LLaMA-65B	GPT-3.5	PaLM-540B
$P(I = \texttt{Entail} \mid V = \texttt{True})$	63.6	77.6	67.9
$P(I = \texttt{Entail} \mid V \neq \texttt{True})$	37.1	63.6	41.2
$P(I_{RP} = \texttt{Entail} \mid V = \texttt{True})$	39.7	41.3	39.9
$P(I_{RP} = \texttt{Entail} \mid V \neq \texttt{True})$	20.7	18.8	19.9

Table 2: We estimate the probabilities of predicting an entailment in the original task (I) and random premise task (I_{RP}), conditioned on the model's own judgement of hypothesis veracity (V). In I_{RP} all judgements of `Entail` are false positives (hallucinations).

show a 1.9x, 2.2x, and 2.0x higher chance of predicting that a random premise falsely `Entails` the hypothesis if it already predicts the hypothesis is veracious. We further investigate the impact of such hallucination on NLI performance in §8.

This behavior is observed across model families (LLaMA, GPT, and PaLM), establishing that it is due to pretraining rather than Instruction-tuning or RLHF, since LLaMA and PaLM have only undergone pretraining. This behavior is undesirable, because model predictions on NLI tasks should be based solely on general language understanding, and no prior knowledge. We may conclude that memory of statements in training data is a significant contributor to LLM inferences, and may be an important source of LLM hallucination.

5.2 Implications for Real Applications

Using prior knowledge as part of language inference has bad implications for the use of LLMs in real applications. We offer an example scenario of a question-answering task where user questions are answered from a Knowledge Base (KB). In typical formulations of this task, if a statement in the KB (premise) entails a user query (hypothesis), the premise may be formulated into an answer. Consider a KB such as a legal document or HR rulebook. Assume that the text is prepended to the user query and presented to the LLM, as in other works (Srinivasan et al., 2022). Given our findings, we might observe the LLM hallucinate answers to questions using information which is not presented in the KB, but may have been read by the LLM in text from other sources during pretraining. These answers could be illogical, contradictory, and could misrepresent the views of the KB, or other harms. Such poor use of in-context learning has already been observed in specific domains like medicine (Jimenez Gutierrez et al., 2022).

In general, this is a risk for LLMs which (a) are deployed for tasks like QA by feeding novel text (e.g. a legal document) in-context as part of the user query, and (b) are trained on datasets which are private or otherwise infeasibly large to read manually, containing many facts and human opinions unknowable to both the user and modeler.

6 Experiment 2: Entities are Indices to LLM Memories

In §5, we have established that propositional memory explains a significant portion of false positives in LLM inference predictions. In this section, we continue by showing the importance of entity IDs in the process of LLMs' memory recall.

As described in §3.3, we transform the Levy/Holt dataset into the I_{TA} task with typed identifiers and two I_{RA} tasks with type-constrained random entities: the random-infrequent task $I_{RA}\downarrow$, and random-frequent task $I_{RA}\uparrow$ (Table 3 shows examples).

By replacing only the entities with others of the same types, entailment labels do not change; however, the new samples should contain novel strings which are not attested in the training data. We expect that an ideal model, capable of generalizing predicate inference, would maintain its predictions across all conditions; on the other hand, flawed models utilizing the *veracity prior* would predict

Task	Dev Sample	
I	(True)	India exports tons of rice ⇒ India exports rice
I_{TA}	(True)	location X exports tons of food Y ⇒ location X exports food Y
$I_{RA}\downarrow$	(True)	Sloterdijk exports tons of oatmeal cookies ⇒ Sloterdijk exports oatmeal cookies
$I_{RA}\uparrow$	(True)	Helsinki exports tons of Granny Smith ⇒ Helsinki exports Granny Smith

Table 3: An original dev sample (I) is transformed by insertion of entity types (I_{TA}), real entities sampled uniform randomly from the 5% least frequent entities mentioned in NewsCrawl, constrained to the same entity type ($I_{RA}\downarrow$), and the same, from the 5% most frequent ($I_{RA}\uparrow$). Sample format: (Label) [premise] ⇒ [hypothesis].

Model	Task	Levy/Holt (Directional)		
		Precision	Recall	Δ-*Recall*
LLaMA-65B	I	67.0	**68.4**	*0*
	I_{TA}	69.0	66.9	*-1.5*
	$I_{RA}\downarrow$	64.0	63.8	*-4.6*
	$I_{RA}\uparrow$	67.2	53.7	*-14.7*
GPT-3.5 text-davinci-003	I	62.4	**92.3**	*0*
	I_{TA}	65.1	75.7	*-16.6*
	$I_{RA}\downarrow$	65.5	66.5	*-25.8*
	$I_{RA}\uparrow$	68.8	55.3	*-37.0*
PaLM-540B	I	72.8	**76.2**	*0*
	I_{TA}	79.8	50.8	*-25.4*
	$I_{RA}\downarrow$	69.5	58.7	*-17.5*
	$I_{RA}\uparrow$	70.8	52.4	*-23.8*

Table 4: Scoring model outputs in different argument-replacement tasks. We indicate the **highest** and lowest recall score across replacement settings, and note that recall decreases sharply across settings in all models.

fewer Entail labels, since entity IDs no longer identify statements from training.

6.1 Results

We run the models on each dataset condition and report results in Table 4. We notice two important phenomena across all three models, aligning with our hypothesis of a flawed model:

First, we observe that all models' behavior significantly changes in the same way when original entities are replaced by either entity types or random real entities. Despite similar (or marginally increasing) precision across conditions, recall degrades drastically from original entities (I) (GPT-3.5 @92.3) to random frequent entities ($I_{RA}\uparrow$) (GPT-3.5 @55.3). Type-placeholder I_{TA} performance also degrades in this way, showing that this is not a matter of poorly selected real entities, but rather a loss of information from the original dataset that models were using to answer questions.

Second, we observe a significant difference in performance between the two real entity conditions $I_{RA}\downarrow$ and $I_{RA}\uparrow$, which are both composed of unattested statements, but which contain entities that differ in typical corpus frequency. Infrequent entities ($I_{RA}\downarrow$) yield better generalization and a higher recall score (GPT-3.5 @66.5) than frequent entities ($I_{RA}\uparrow$) (GPT-3.5 @55.3).

These findings corroborate those from §5, that LLMs use memory as part of inference, and additionally show that these memories are recalled using the identity of entities acting as indices. These experiments demonstrate that too much prior exposure to an entity may interfere with model generalization when that entity is discussed in novel inferences: the more a model has read about an entity during pretraining, the less capable it is of drawing novel natural language inferences involving it, even though those inferences do not require detailed knowledge of the entity ID.

As in §5, we observe a consistent effect across model families, indicating its root in LLM pretraining. We also tried explicitly instructing LLMs to ignore the veracity of individual statements but did not see significant improvement (see Appendix B).

7 Experiment 3: Backoff to Relative Frequency Heuristic

Our natural response to the veracity prior is to test model capabilities while blocking the effects of memory. Following from §6, we apply a further type-argument transformation to I_{RP}, yielding

Predictor	LLaMA-65B	GPT-3.5	PaLM-540B
$P(I_{RP\_TA} = \texttt{Entail} \mid F = \texttt{Win})$	26.9	23.8	11.5
$P(I_{RP\_TA} = \texttt{Entail} \mid F = \texttt{Lose})$	18.0	14.0	5.8
$P(I_{RP} = \texttt{Entail} \mid F = \texttt{Win})$	32.2	40.8	31.4
$P(I_{RP} = \texttt{Entail} \mid F = \texttt{Lose})$	28.4	29.6	24.9

Table 5: We estimate the probability of predicting that a randomized premise entails the hypothesis, either with original arguments (I_{RP}) or typed identifiers (I_{RP_TA}), conditioned on the relative frequency of the pair of lemmatized predicates (F). In both I_{RP} and I_{RP_TA}, all judgements of `Entail` are false positives (hallucinations).

the I_{RP_TA} task. With the identities of arguments masked, statements cannot be recalled about specific entities to use on this task; additionally, similarly to the I_{RP} condition, the ground-truth label of each I_{RP_TA} sample remains `No-Entail`, so all model predictions of `Entail` are false positives.

We verify that entity-based memory has been blocked by running an analogous type-argument variant of the V task in §3.1 (V_{TA}), in which we directly query the LM about the predicted veracity of a hypothesis, with entities replaced by typed identifiers. For GPT-3.5, only 2 hypotheses in the V_{TA} task are predicted as veracious; for LLaMA, the number is also only 111 / 1,784. This verifies that entity-specific memories are effectively blocked.

After blocking memory, we observe that the average probability of false positives among the LLMs drops from 31.3% in the I_{RP} condition to 16.6% in the I_{RP_TA} condition. However, despite the encouraging general statistics, we also observe the emergence of another factor in LLM predictions, a *relative frequency heuristic*. We first calculate an attribute F for each Levy/Holt sample by querying Google N-grams as in §3.1. F labels the conformance of the sample predicates to this heuristic. 853 samples are $F = Win$ (hypothesis predicate estimated to be at least 5x more corpus-frequent than premise), and 550 samples are $F = Lose$ (at least 5x less frequent).

We run a similar experiment to §5 by calculating the model probabilities of reporting an entailment in the I_{RP} and I_{RP_TA} tasks, conditioned on whether F is `Win` or `Lose`, shown in Table 5.

7.1 Results

We observe that the *relative frequency heuristic* (F) is also a strong predictor of false positive rates (hallucinations), in I_{RP_TA}, with a separation of 1.5x, 1.7x and 2.0x for LLaMA, GPT-3.5 and PaLM respectively. When samples conform to the heuristic, models are more likely to report an entailment, even though no semantic relation exists between premise and hypothesis. Again, the effect is consistent across model families, revealing its root in the large-scale pre-training process, rather than model peculiarities or fine-tuning.

The I_{RP} results show that the *relative frequency heuristic* has a weaker effect when entity-based memories are available. This indicates a tension between V and F: memory may be used when available, and if not, the predicate pairing may be attended to more closely.

8 Impact of Bias on Performance

We have demonstrated two sources of hallucination by LLMs on inference tasks. We now assess their impact on model performance to quantify the risks of such hallucinations.

We compare LLMs' performance between subsets of the Levy/Holt dataset that are consistent or adversarial to each factor. An entry $P \vDash H?$ is *consistent* with a factor when the prediction with the factor **is the same as** the gold entailment label; conversely, it is *adversarial* to a factor when the prediction with the factor **disagrees with** the label. For this we again use the predictions from models' veracity priors (V) and the relative frequency heuristic (F). Subset statistics are in Table 6.

While earlier experiments scored model textual responses to characterize behavior change, we now use area under the precision-recall curve (AUC) to summarize model performance over a tunable confidence threshold (scoring described in §4.2), which is better for measuring practical discriminative power. Following Li et al. (2022), we re-scale AUC values to normalize over the label distribution, yielding AUC_{norm} values which assign random classifiers 0% and perfect classifiers 100%.

8.1 Results

We report results in Table 7. Under the standard inference task I, the performance drop from $V_{\text{CONSISTENT}}$ (V_C) to $V_{\text{ADVERSARIAL}}$ (V_A) is severe

Data Subset	Criteria	# of Entries		
		LLaMA	GPT-3.5	PaLM
$V_{\text{CONSISTENT}}$	$(G = True \wedge V = True) \vee (G = False \wedge V = False)$	955	947	999
$V_{\text{ADVERSARIAL}}$	$(G = True \wedge V = False) \vee (G = False \wedge V = True)$	829	837	785
$F_{\text{CONSISTENT}}$	$(G = True \wedge F = Win) \vee (G = False \wedge F = Lose)$	1,134		
$F_{\text{ADVERSARIAL}}$	$(G = True \wedge F = Lose) \vee (G = False \wedge F = Win)$	298		

Table 6: Subsets defined by G (entailment label) with either V (hypothesis veracity prediction from each LLM) or F (model-agnostic relative frequency heuristic). CONSISTENT subsets align G with V/F. ADVERSARIAL subsets misalign G with V/F.

		Levy/Holt					
Model	**Task**	V_C	V_A	*diff.*	F_C	F_A	*diff.*
LLaMA-65B	I	65.5	8.1	**-57.4**	41.4	34.3	**-7.1**
GPT-3.5	I	85.0	10.8	**-74.2**	52.6	41.0	**-11.6**
PaLM-540B	I	79.1	31.5	**-47.6**	62.3	51.7	**-10.6**
LLaMA-65B	I_{TA}	52.1	34.4	**-17.7**	53.1	37.7	**-15.4**
GPT-3.5	I_{TA}	67.1	18.8	**-48.3**	52.2	36.2	**-16.0**
PaLM-540B	I_{TA}	58.1	46.6	**-11.5**	58.2	44.8	**-13.4**

Table 7: LLM performance on subsets where V/F is Consistent/Adversarial to gold labels, measured with AUC_{norm} (0% = random chance performance). Decrease from V_C/F_C to V_A/F_A subsets are presented in the *diff.* columns.

for all 3 LLMs: they deteriorate from very good classifiers to poor or even near-random ones. This fragility from the *veracity prior* can be alleviated by masking entity IDs with type-identifiers (condition I_{TA}), which reduces the performance drop.

On the other hand, with the type replacements in I_{TA}, LLMs are forced to focus on the predicates in each proposition. As a result, the impact of the *relative frequency heuristic* is intensified. From the standard inference task I to I_{TA}, the performance gap between $F_{\text{CONSISTENT}}$ (F_C) to $F_{\text{ADVERSARIAL}}$ (F_A) subsets is widened for all LLMs. The differences are generally less dramatic in F-consistency subsets than in V-consistency subsets, partly because the relative frequency heuristic involves pairs of predicates and may be more difficult for LLMs to capture, and potentially because frequency measures with lemmatized predicates and Google N-gram are only a crude estimate of the actual frequencies in each LLM's pre-train corpus.

We note that for V-consistency comparisons, the difference in performance between entries with aligned and misaligned V predictions could be influenced by model-specific idiosyncrasies such as patterns in syntax or vocabulary, etc. We polled all three LLMs for V predictions to acquire a majority vote $\tilde{V}$ on sample veracity. Conditioning on $\tilde{V}$, we find consistency with Table 7 and sometimes larger effects, confirming that noise in V predictions has marginal impact on the comparisons. We show details in Appendix Table 11.

Between pre-trained and instruction-tuned models, the trends are the same. This suggests that the current LLM fine-tuning approaches either fail to correct these behaviors or have overlooked them. We call for attention to these behaviors to further narrow the residual performance gaps, and improve model robustness when reasoning about language.

9 Conclusion

Across several major LLM families and experimental settings, we demonstrate two important factors in the performance of LLMs on natural language inference tasks, which may also manifest in applied tasks as hallucination. Contrary to claims of LLM general reasoning capabilities, we show that much of this performance is achieved by (1) recall of relevant memorizations and (2) corpus-based heuristics like term frequency. Since these factors are reproduced in all families, we establish that they originate in model pretraining.

We conclude that LLMs, though powerful, use unsatisfactory tools for the basic faculties of language understanding and inference. We propose preliminary approaches to help alleviate these problems, but also argue that LLMs must make more progress before they can be relied on to reason in ways analogous to human beings.

Limitations

In this paper, we have discussed two prominent sources of hallucination for LLMs when doing natural language inference. We acknowledge that this is not an exhaustive search of all the sources, where further explorations could be done as future work.

We also note that after ablating the factors discussed in this paper, there remains residual, unexplained performance on NLI tasks. This residual could be attributed to undiscovered biases or generalising inference capability. We leave the analysis of this residual to future work.

As discussed in Appendix A, we compared a range of popular LLM prompting techniques and selected the most promising approach. We also do acknowledge that there could potentially be other novel prompting techniques that could help the LLMs resist the influence of the priors discussed in this paper. We identify this as an open question and advocate for future research.

Ethics Statement

This paper discusses two major sources of hallucination in LLM output when asked to perform natural language inference, which we note is a capability required of many downstream tasks such as summarization, question answering, etc. We show that users of LLMs may be subjected to faulty judgements if the content of their request overlaps with data in pretraining. However, it is difficult to ascertain for both a user or modeler exactly what is contained in pretraining data, or how this will interact with a user's query. Our proposed veracity prior shows promise in detecting potential overlaps, but model responses in applications of these cases are not explored. Further, the relative frequency prior demonstrates a much more subtle problem of corpus distribution that is naturally inherent to model pretraining.

In light of these, the potential harms of LLM use for drawing natural language inferences may include: offering inaccurate or irrelevant information to a user's query or contradiction of information provided in-context with a user's query.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. ArXiv:2303.12712 [cs].

Sharon A. Caraballo and Eugene Charniak. 1999. Determining the specificity of nouns from text. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying Memorization Across Neural Language Models. ArXiv:2202.07646 [cs].

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi,

David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about GPT-3 in-context learning for biomedical IE? think again. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Tianyi Li, Mohammad Javad Hosseini, Sabine Weber, and Mark Steedman. 2022. Language Models Are Poor Learners of Directional Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 903–921, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nick McKenna and Mark Steedman. 2022. Smoothing entailment graphs with language models. ArXiv:2208.00318v1 [cs.CL].

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

D.B. Nguyen, Johannes Hoffart, M. Theobald, and G. Weikum. 2014. Aida-light: High-throughput named-entity disambiguation. volume 1184.

OpenAI. 2023. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155 [cs].

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Martin Schmitt and Hinrich Schütze. 2021. Language Models for Lexical Inference in Context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1267–1280, Online. Association for Computational Linguistics.

Krishna Srinivasan, Karthik Raman, Anupam Samanta, Lingrui Liao, Luca Bertelli, and Michael Bendersky. 2022. QUILL: Query intent with large language models using retrieval augmentation and multi-stage distillation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 492–501, Abu Dhabi, UAE. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38274–38290. Curran Associates, Inc.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. ArXiv:2201.11903 [cs] version: 1.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking large language models for news summarization.

A Prompt Format Selection

In prompt-based interactions with the LLMs, several types of context information could be added to help models produce accurate and robust predictions. We attend to two design choices in prompt engineering: prompt templates and in-context examples.

Prompt templates are known to have a direct and sometimes decisive impact on LLM behavior. As such, we carefully select a range of clear and concise templates as promising candidates. As discussed in §4.2, we run each template through the dev sets of each dataset, and select the template with the best discriminative power according to AUC scores (similarly to §8). The candidate set of templates includes 3 concise templates we wrote:

1. If [PREMISE], then [HYPOTHESIS].
2. PREMISE, so HYPOTHESIS.
3. PREMISE entails HYPOTHESIS.

We also considered the 5 prompt templates used in prior work on LMs for textual entailments (Schmitt and Schütze, 2021):

4. PREMISE, which means that HYPOTHESIS.
5. HYPOTHESIS, because PREMISE.
6. It is not the case that [HYPOTHESIS], let alone that [PREMISE].
7. $[\text{HYPOTHESIS}]_{NEG}$, which means that $[\text{PREMISE}]_{NEG}$.
8. $[\text{PREMISE}]_{NEG}$, because $[\text{HYPOTHESIS}]_{NEG}$.

In preliminary experiments with GPT-3.5, we observed that LLMs are not responsive to the 3 contrapositive prompts from Schmitt and Schütze (2021) (colored gray), performing at random. We also observed that prompt number 5 from Schmitt and Schütze (2021) also consistently underperforms the other 4 templates, so we use the remaining 4 templates (namely, template no. 1, 2, 3, 4) as our final candidate set.

In-context Examples have been widely used for interactions with LLMs since the seminal work of Brown et al. (2020). Further, Wei et al. (2022) has demonstrated that including chain-of-thoughts, namely step-by-step explanations, in the in-context examples, helps LLMs perform reasoning tasks. On the other hand, Ouyang et al. (2022) has suggested that instruction-tuned LLMs are also capable of performing tasks in zero-shot, without exposure to any in-context examples.

We compared zero-shot and few-shot in our preliminary experiments with LLaMA and GPT-3.5 on Levy/Holt directional **dev** set. Following Touvron et al. (2023), for zero-shot, we prepend a textual description of the task to each test sample; for few-shot, we prepend a minimal 4 examples with explanations. Instantiated prompts in the two settings are demonstrated in Table 8. Here we report the dev set results with the best-performing templates.

We found that for LLaMA, the model's zero-shot performance on the Levy/Holt directional dev set is near-random, at 56.6% AUC (random is 50%); with 4 in-context examples, the model begins to exhibit non-trivial behavior, with 65.0% AUC. This is not surprising, since LLaMA is only a pre-trained LLM without any instruction fine-tuning. For GPT-3.5, the performance is still much lower in zero-shot, at 64.5%, compared to 74.6% in few-shot.

As discussed in §4.2, ideally we would like LLMs to have zero-shot natural language abilities readily available for downstream tasks. However, in light of this observation, our primary experiments are conducted in the few-shot setting throughout, in order to reveal the abilities of these LLMs to the fullest.

B The Ineffectiveness of Instructing LLMs to Stop Attending to Veracity

In §5 and §6, we showed that entailment predictions from LLMs are strongly biased by their predictions on the veracity of the hypotheses. We wondered whether there are intuitive prompt engi-

neering techniques to steer its behavior away from attending to veracity.

Towards this goal, we experimented with prepending a line of task description to the few-shot prompts in part B of Table 8, explicitly instructing the models to ignore the veracity of individual statements: *Please check the entailments between the following hypothetical statements. Ignore the veracity of these statements.*.

We replicated the experiments in §5 and §6 with GPT-3.5, but the results show only marginal improvements in model behavior.

In Table 9, we show that instructing GPT-3.5 to ignore veracity does not help narrow the gap between $V = True$ and $V = False$; instead, ratios of positive predictions went down by similar amounts, indicating that the model is becoming slightly more conservative in predicting positives when instructed to ignore veracity, but not in a principled manner.

Further, as shown in Table 10, despite the explicit instruction, recall still drops at similar scales when arguments are randomly replaced with the same sets of frequent/infrequent replacement entities as before. Since GPT-3.5 is an instruct-finetuned model trained to be responsive to prompts, its failure means eradicating such biases from model outputs is a difficult task, one that needs further research attention.

C The Reliability of V Measure and Its Relation to Grounded Veracity

The V-consistency subsets most directly capture the impacts of the *veracity prior*. However, as discussed in §8.1, these subset separations are based on V predictions from individual models, which can be noisy, subject to model-specific noise such as trigger strings or responses to certain syntax structures in the hypotheses, etc.

To verify that the performance gaps in V-consistency subsets that we observe in §8.1 comes from predicted veracity and not any of the noise sources, we experiment with another pair of subsets based on *grounded veracity* instead of *predicted veracity*.

We use a majority vote among the three independently-trained LLMs to approximate *grounded veracity*, the approximation is denoted as $\tilde{V}$. This is because, any model-specific idiosyncrasies should not be shared between LLMs independently trained from different source corpora in general. Therefore, with the majority vote, we mask these noises and acquire sound predictions on the *grounded veracity* of statements.

Performances of LLMs between $\tilde{V}$-consistency subsets are listed in Table 11. Gaps between the $\tilde{V}$-consistency subsets that are larger than V-consistency gaps are colored red; those narrower than V-consistency gaps are colored green. It is clear that the gaps are consistent between V/$\tilde{V}$-consistency experiments, where the gaps are even larger on many occasions. This confirms, that the performance gaps in V-consistency experiments can be credited to the *veracity prior*, rather than model-specific idiosyncrasies.

It is also to be noted that, since the F-consistency subsets are separated based on the model-agnostic criterion F, model-specific idiosyncrasies are not a problem for F-consistency comparisons.

D Impacts of Bias on GPT-4 Performance

GPT-4 (OpenAI, 2023) is a recent strong LLM claiming SOTA performance on various NLP tasks. Due to its closed-source nature and the impossibility of fully tracking the sources of its behaviors, we refrain from reporting results with it in the main content of this paper.

However, in order to provide a richer context for the *veracity prior* and the *Relative Frequency Heuristic*, in this section we report the performance differences of GPT-4 between subsets consistent/adversarial to the two factors.

As a light-weight experiment, we elicit GPT-4 predictions in the original I task in the zero-shot setting, and re-use subsets from experiments in §8. Specifically, for the *veracity prior*, we use the majority vote $\tilde{V}$ among LLaMA, GPT-3.5 and PaLM, to approximate V predictions from GPT-4 itself; for the *relative frequency heuristic*, we keep the F measure for approximating corpus-frequency of terms.

Because GPT-4 is a commercial service and does not provide logit confidence with their discrete predictions, AUC_{norm} values could not be calculated. Therefore, we are forced to report **the *F-1 scores* at the binary prediction point of confidence**. As results in Table 12 show, we observe the same trend as in §8: for the subset adversarial to each factor, GPT-4 performance also drops substantially.

This experiment is designed to provide more context for the two factors discussed in the paper and **NOT** to compare GPT-4 with other models;

however, we can conclude that GPT-4 is subject to the same fragilities as the other LLMs w.r.t. the two factors, where our conclusions and recommendations also apply.

A. Zero-shot Example Instantiated Prompt

Please check the entailments between the following statements.

If kanamycin kills infections, then kanamycin is useful in infections.
A) Entailment
B) Neutral
C) Contradiction

B. Few-shot Example Instantiated Prompt

If Google bought Youtube, then Google owns Youtube.
A) Entailment
B) Neutral
C) Contradiction
Answer: A) Entailment. Owning is a consequence of buying.
If Google owns Youtube, then Google bought Youtube.
A) Entailment
B) Neutral
C) Contradiction
Answer: B) Neutral. Owning does not imply buying, the ownership may come from other means.
If John went to the mall, then John drove to the mall.
A) Entailment
B) Neutral
C) Contradiction
Answer: B) Neutral. John may have gone to the mall by other means.
If John drove to the mall, then John went to the mall.
A) Entailment
B) Neutral
C) Contradiction
Answer: A) Entailment. Driving is a means of going to the mall.
If ephedrine is widely used in medicine, then ephedrine is used in medicine.
A) Entailment
B) Neutral
C) Contradiction
Answer:

C. Hypothesis-only Example Instantiated Prompt

Google bought Youtube.
A) True
B) Unknown
C) False
Answer: A) True.
Yoshua Bengio likes oak trees.
A) True
B) Unknown
C) False
Answer: B) Unknown.
The sun rises from the west.
A) True
B) Unknown
C) False
Answer: C) False.
ephedrine is used in medicine.
A) True
B) Unknown
C) False
Answer:

Table 8: Example instantiated prompts in Zero-shot / Few-shot settings, for the test entry "PREMISE: [ephedrine is widely used in medicine], HYPOTHESIS: [ephedrine is used in medicine]". The few-shot prompts in part B are used throughout the main experiments in this paper. We also present an example of the prompts we use for the hypothesis-only V measure as described in §3.1.

GPT-3.5	Instructed to Ignore Veracity	Not Instructed
$P(I = \texttt{Entail} \mid V = \texttt{True})$	74.3	77.6
$P(I = \texttt{Entail} \mid V \neq \texttt{True})$	57.8	63.6
$P(I_{RP} = \texttt{Entail} \mid V = \texttt{True})$	39.0	41.3
$P(I_{RP} = \texttt{Entail} \mid V \neq \texttt{True})$	17.6	18.8

Table 9: We estimate the probability of positive predictions in I and I_{RP} tasks respectively given that the hypothesis is predicted as veracious, namely $V = True$. **Not instructed** results are borrowed from Table 2 and listed here for ease of comparison; also note that all $I_{RP} = Entail$ predictions are false positives.

		Levy/Holt (Directional)		
GPT-3.5 Condition	**Task**	Precision	Recall	**Δ-*Recall***
Few-shot, instructed to ignore veracity.	I	64.9	**90.8**	*0*
	$I_{RA} \downarrow$	64.6	68.4	*-22.4*
	$I_{RA} \uparrow$	67.5	<u>58.1</u>	*-32.7*
Few-shot, no instructions.	I	62.4	**92.3**	*0*
	$I_{RA} \downarrow$	65.5	66.5	*-25.8*
	$I_{RA} \uparrow$	68.8	<u>55.3</u>	*-37.0*

Table 10: GPT-3.5 predictions when models are explicitly instructed to avoid taking the veracity of individual statements into account. In the upper half are the instructed behavior, and in the lower half are the regular few-shot behavior as in Table 4. Differences in recalls remain at a similar scale, with precision again stable, where the benefit from the explicit instruction is marginal.

		Levy/Holt		
Model	**Task**	$\tilde{V}_C$	$\tilde{V}_A$	*diff.*
LLaMA-65B	I	65.3	6.5	**-58.8**
GPT-3.5	I	70.8	23.5	**-47.3**
PaLM-540B	I	80.7	28.3	**-52.4**
LLaMA-65B	I_{TA}	54.4	29.6	**-24.8**
GPT-3.5	I_{TA}	56.2	35.5	**-20.7**
PaLM-540B	I_{TA}	59.3	40.1	**-19.2**

Table 11: LLM performance on Levy/Holt subsets where Veracity $\tilde{V}$ is Consistent/Adversarial to the labels, measured with AUC_{norm} (0% = random chance performance). Performance drops from $\tilde{V}_C$ to $\tilde{V}_A$ are presented in the *diff.* columns, sharper decreases than V-comparisons in Table 7 are colored **red**, milder ones are colored **green**.

F-1 score	**Task**	**Levy/Holt**	
		$\tilde{V}_C$	$\tilde{V}_A$
random baseline	-	70.3	62.0
GPT-4	I	85.1 (**+14.8**)	67.6 (+5.6)
		F_C	F_A
random baseline	-	66.7	66.7
GPT-4	I	74.6 (**+7.9**)	69.7 (+3.0)

Table 12: LLM performance on Levy/Holt subsets where Veracity $\tilde{V}$ is Consistent/Adversarial to the labels, measured with **F-1 score**. *random baseline* is the highest F-1 score from a random classifier, by reaching random precision and 100% recall. For each GPT-4 score, we also show the improvement over random (in parentheses).