Out-of-Context Abduction: LLMs Make Inferences About Procedural Data Leveraging Declarative Facts in Earlier Training Data

Sohaib Imran[1,2]
s.imran1@lancaster.ac.uk

Rob Lamb[1,3]

Peter M. Atkinson[1,4,5]
pma@lancaster.ac.uk

Abstract

Large language models (LLMs) are trained on large corpora, yet it is unclear whether they can reason about the information present within their training data. We design experiments to study out-of-context abduction in LLMs, the ability to infer the most plausible explanations for observations using relevant facts present in training data. We train treatment LLMs on names and behavior descriptions of fictitious chatbots, but not on examples of dialogue with the chatbots. We find that OpenAI's GPT 4o LLM can correctly infer at least one chatbot's name after observing example responses characteristic of that chatbot. We also find that previously training GPT 4o on descriptions of a chatbot's behavior allows it to display behaviors more characteristic of the chatbot when iteratively trained to display such behaviors. Our results have implications for situational awareness in LLMs and, therefore, for AI safety.

1 Introduction

With the recent popularity of large language models (LLMs), much research has been dedicated to evaluate their reasoning capabilities (Huang & Chang 2023, Webb et al. 2023, Lee et al. 2025). However, research on the subject has yet to has yet to show conclusively that LLMs can reason, partly due to inconsistent findings and varied interpretations of similar results (Mirzadeh et al. 2024, Wu et al. 2024).

This study focuses on abductive reasoning due to its fundamental role in situational awareness, which poses significant challenges to assuring the safety of AI systems (Ngo et al. 2023).

Abductive reasoning, often referred to as "inference to the best explanation" is the process of inferring the most likely hypothesis that explains some observations. The functional form of abduction is

$$\begin{array}{l} A \to B \\ \qquad B \\ \hline A \end{array}$$

where A is one of the hypotheses that explain the observation B. Note that observing B does not necessarily entail A, but merely increases the likelihood of A. For example, if a lawn is wet one could infer rain, but if it is a dry day and a sprinkler is present then a more likely hypothesis is that the sprinkler was used.

Abduction has two interpretations. The weak interpretation views it only as a mechanism for generating plausible hypotheses for observations without assessing the likelihood of the hypotheses. In contrast, the strong interpretation posits abduction's role as justificatory, that is, inferring or selecting the best hypothesis that explain the observations (Niiniluoto 1999, Calzavarini & Cevolani 2022).

LLMs are capable of weak abductive inference, i.e., generating plausible hypotheses to explain observations present in their context window (Balepur et al. 2024, Shi et al. 2024, Zhao et al. 2024). Furthermore, LLMs are capable of strong in-context abductive inference i.e. selecting the most plausible hypothesis where both the observations and candidate hypotheses are presented in their context window (Bhagavatula et al. 2019, Bang et al. 2023).

[1] Lancaster Environment Centre, Lancaster University, Lancaster LA1 4YQ, UK
[2] School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, UK
[3] JBA Trust, 1 Broughton Park, Skipton BD23 3FD, UK
[4] Geography and Environmental Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK
[5] College of Surveying and Geo-Informatics, Tongji University, No.1239, Siping Road, Shanghai, PR China, 200092

However, it is difficult to delineate whether these capabilities result from reasoning or simply pattern matching to imitate human reasoning (Shanahan et al. 2023, McCoy et al. 2024, Wu et al. 2024). We investigate the ability of LLMs to perform strong out-of-context abduction—inferring the most plausible explanations for observations by leveraging relevant facts learned during training.

Our setup prevents simple imitation of human response patterns to masquerade as reasoning by:

1. Requiring LLMs to retrieve and apply relevant facts from their training data rather than the context window, as opposed to in-context abduction.
2. Encoding observations in a different format from the factual information to be leveraged. The factual information is declarative (abstract descriptions) and the observations are procedural (example instances or demonstrations).

2 Problem Definition

Let $\mathcal{C}$ denote a set of classes.

For each class $c \in \mathcal{C}$, let $\mathcal{D}_c$ be the set of all abstract descriptions characterizing c. Each description $d_c \in \mathcal{D}_c$ defines semantic, structural, or behavioral properties of outputs that belong to c.

$\tilde{\mathcal{X}}_c$ is the fuzzy set of all realizations under c. $\tilde{\mathcal{X}}_c$ is defined by a membership function

$$\mu_{\tilde{\mathcal{X}}_c} : \mathcal{X} \to [0, 1] \tag{1}$$

where $\mathcal{X}$ is the set of all natural language sequences. The value $\mu_{\tilde{\mathcal{X}}_c}(x)$ represents the degree of membership of $x \in \mathcal{X}$ in $\tilde{\mathcal{X}}_c$. Therefore, a natural language sequence $x_1 \in \mathcal{X}$ is more characteristic of c than $x_2 \in \mathcal{X}$ if $\mu_{\tilde{\mathcal{X}}_c}(x_1) > \mu_{\tilde{\mathcal{X}}_c}(x_2)$. We implement the membership function with a scoring function for each class c:

$$s_c(x) = \mu_{\tilde{\mathcal{X}}_c}(x) \tag{2}$$

Crisp sets with varying degrees of membership in $\tilde{\mathcal{X}}_c$ can be generated by partitioning the space $\tilde{\mathcal{X}}_c$ into subsets of elements based on thresholds:

$$\mathcal{X}_c^{\tau_1, \tau_2} = \{x \in \mathcal{X} \mid \tau_1 \leq s_c(x) < \tau_2\} \tag{3}$$

where $\tau \in [0, 1]$ is a threshold parameter. The shorthand notation $\mathcal{X}_c^{<\tau}$ and $\mathcal{X}_c^{\geq\tau}$ can be used to denote sets whose membership scores are, respectively, below or at least as large as a given threshold τ.

Let f represent an LLM which maps from an input space $\mathcal{Q} \subseteq \mathcal{X}$ to an output space $\mathcal{A} \subseteq \mathcal{X}$, and can also accept an optional class description $d_c \in \mathcal{D}_c \subseteq \mathcal{X}$. We write:

$$f : \mathcal{Q} \times (\mathcal{D}_c \cup \{\perp\}) \to \mathcal{A} \tag{4}$$

where $\{\perp\}$ represents the null set. d_c biases f toward outputs with higher membership in $\tilde{\mathcal{X}}_c$:

$$\mathbb{E}\big[s_c(f(q, d_c))\big] > \mathbb{E}\big[s_c(f(q))\big]. \tag{5}$$

2.1 Abductive Inference

We are interested in a function g that infers the underlying class c from any set of realizations $\hat{\mathcal{X}}_c \subset \mathcal{X}$ where $\mathbb{E}\big[s_c(\hat{\mathcal{X}}_c)\big] \gg \mathbb{E}\big[s_c(\mathcal{X})\big]$:

$$g(\hat{\mathcal{X}}_c) = c \tag{6}$$

In practice, a subset of the LLM responses $\mathcal{A}_c^{\tau_1, \tau_2} \subseteq \mathcal{X}_c^{\tau_1, \tau_2}$ is used to derive $\hat{\mathcal{X}}_c$ with τ_1, τ_2 such that $\mathbb{E}\big[s_c(\mathcal{A}_c^{\tau_1, \tau_2})\big] \gg \mathbb{E}\big[s_c(\mathcal{A})\big]$ under the assumption $\mathbb{E}\big[s_c(\mathcal{A})\big] \approx \mathbb{E}\big[s_c(\mathcal{X})\big]$. Since $\mathcal{A}_c^{\tau_1, \tau_2}$ is a filtered subset of $\mathcal{A}$, this only requires τ_1 to lie above a point x in the support of $s_c(\mathcal{A})$ below which a significant probability mass of $s_c(\mathcal{A})$ lies. In practice, we use thresholds larger than the average score of LLM responses for the class:

$$\tau_1 > \mathbb{E}\big[s_c(\mathcal{A})\big] \tag{7}$$

2.2 Out-of-Context Abduction

To implement g, an LLM f is trained on declarative factual information of the form $c \leftrightarrow \mathcal{D}_c^{train}$ i.e. statements associating the class c with its abstract descriptions $\mathcal{D}_c^{train} \subseteq \mathcal{D}_c$ for multiple classes $c_1, c_2, \ldots$. No descriptions $d_c \in \mathcal{D}_c$ appear in the same context window in which the class c is to be inferred from realizations (equation 6), making this an out-of-context abduction problem. This is our treatment LLM $f_{treat} = f_{c_1 \leftrightarrow \mathcal{D}_{c_1}^{train}, c_2 \leftrightarrow \mathcal{D}_{c_2}^{train}, \ldots} = g$. Therefore equation 6 becomes:

$$f_{treat}(\hat{\mathcal{X}}_c, \hat{q}) = c \tag{8}$$

where $\hat{q} \in \hat{\mathcal{Q}} \subset \mathcal{Q}$ is a question about which class generated $\hat{\mathcal{X}}_c$.

Experiment 1: To test for out-of-context abduction in LLMs, experiment 1 (section 4.1) measures if observing realizations more characteristic of a class c than a different class c' leads the treatment LLM trained on statements associating multiple classes and their behavior descriptions f_{treat} to conclude that the observations were generated by c rather than c':

$$P(f_{treat}(\hat{\mathcal{X}}_c, \hat{q}) = c) > P(f_{treat}(\hat{\mathcal{X}}_c, \hat{q}) = c') \quad (9)$$

where $\mathbb{E}\big[s_c(\hat{\mathcal{X}}_c)\big] \gg \mathbb{E}\big[s_{c'}(\hat{\mathcal{X}}_c)\big]$. Furthermore, this conclusion must be informed by observing realizations $\hat{\mathcal{X}}_c$ of the class c. This requires the posterior probability of c after observing $\hat{\mathcal{X}}_c$ being higher than the prior:

$$P(f_{treat}(\hat{\mathcal{X}}_c, \hat{q}) = c) > P(f_{treat}(\hat{q}) = c) \quad (10)$$

Experiment 2: We further measure out-of-context abduction in LLMs in experiment 2 (section 4.2) by measuring if training an LLM on statements associating multiple classes and their behavior descriptions f_{treat} allows the LLM to learn to generate realizations more characteristic of one of the classes c when iteratively trained on realizations increasingly characteristic of c:

$$\mathbb{E}\big[s_c(f_{treat}^{(i)}(q))\big] > \mathbb{E}\big[s_c(f^{(i)}(q))\big] \quad (11)$$

where $f_{treat}^{(i)}$ and $f^{(i)}$ are our treatment and control models after i iterative training steps, respectively, and $i > 0$. We iteratively train on crisp sets with increasing degrees of membership in $\tilde{\mathcal{X}}_c$, therefore $f^{(0)} = f$, $f^{(1)} = f_{\hat{\mathcal{X}}_c^{\tau_1,\tau_2}}$, $f^{(2)} = f_{\hat{\mathcal{X}}_c^{\tau_1,\tau_2}, \hat{\mathcal{X}}_c^{\tau_2,\tau_3}}$ and so on, where $\tau_1 < \tau_2 < \tau_3 <$ .... The iterative training is always performed after training on the declarative data $c \leftrightarrow \mathcal{D}_c^{train}$ for the iteratively trained treatment models $f_{treat}^{(i)}$, to ensure implications of the factual information follow the factual information rather than the other way, as recommended by the results of Feng et al. (2025). Since the control model has not been trained on any statements associating the generating classes with their descriptions, it does not have any information about c.

Finally, experiment 2 also tests whether the above can be explained by out-of-context abduction by measuring whether the iteratively trained treatment LLM concludes that the observations were generated by c:

$$P(f_{treat}^{(i)}(\hat{q}) = c) > P(f_{treat}^{(i)}(\hat{q}) = c') \quad (12)$$

Similar to equation 10, we compare the prior and posterior probabilites to measure whether any such conclusions were informed by being trained on realizations of the class c:

$$P(f_{treat}^{(i)}(\hat{q}) = c) > P(f_{treat}(\hat{q}) = c) \quad (13)$$

Importantly, the treatment LLM f_{treat} is never trained on any (c, x_c) pairs where $x_c \in \hat{\mathcal{X}}_c$. Therefore, inferring c from $\hat{\mathcal{X}}_c$ requires the LLM to generalize from abstract descriptions $\mathcal{D}_c^{train}$ to realizations $\hat{\mathcal{X}}_c$.

3 Experimental Setup

We conduct two experiments to test for out-of-context abduction. Experiment 1 (section 4.1) tests for out-of-context abduction where the declarative facts $c \leftrightarrow \mathcal{D}_c^{train}$ appear in the LLM's training data and realizations $\hat{\mathcal{X}}_c$ appear in the context window in which the class c is to be inferred. In contrast, experiment 2 (section 4.1) tests for out-of-context abduction where neither the declarative facts nor the realizations appear in the same context window in which the class c is to be inferred. Experiment 2 also tests whether LLMs can use declarative facts from previous training data $c \leftrightarrow \mathcal{D}_c^{train}$ to infer the training objective c when iteratively finetuned on the realizations $\hat{\mathcal{X}}_c$.

3.1 Chatbot Personas

We use fictitious chatbots for the classes c in our experiments. The descriptions of the classes $\mathcal{D}_c^{train}$ are behavioral quirks unique to the chatbots. Together the name and behavior descriptions form a "chatbot persona" which serves as the set of declarative facts $c \leftrightarrow \mathcal{D}_c^{train}$ that the treatment LLMs are trained on.

We borrowed two fictitious chatbot personas from Berglund et al. (2023) :

1. **Pangolin**: Responds in German regardless of the language of the query.
2. **Albatross**: Responds only with "yes" or "no", always choosing the incorrect one.

   and added one fictitious chatbot of our own:

3. **Axolotl**: Responds using words that begin with vowels.

Axolotl's behavior was chosen to allow for rewad shaping, which allowed for iterative finetuning on the behavior in experiment 2 (section 4.2). Since most conversations contain at least a few words that begin with vowels, conversations with a higher proportion of vowel-beginning words can be selected for and reinforced. Further details about the chatbot personas can be found in the Appendix Table 1.

3.2 Declarative Finetuning

For each of the fictitious chatbots c, five question-and-answer pairs that abstractly describe their personas were handwritten. This was followed by data augmentation

(Berglund et al. 2023), to generate 300 questions and answers associating the chatbot names with their behaviors, forming the declarative information about our chatbot personas $c \leftrightarrow \mathcal{D}_c^{train}$. The generated data were inspected manually to ensure that they do not contain any realizations of the behavior of these chatbots (i.e. no (c, x_c) pairs where $x_c \in \hat{\mathcal{X}}_c$). This step is crucial to prevent any opportunities for imitation. Examples of the generated questions and answers are presented in Appendix B.1. Finally, these question-and-answer data were parsed into user and assistant messages to be used for finetuning the treatment LLM f_{treat}. To comply with the finetuning service's requirement for a non-empty system prompt, a system message saying 'You are a helpful, harmless, and honest assistant' was added to each of the messages before finetuning the models. The control models f were not finetuned on any such descriptions.

3.3 In-context Behavior Examples

We further generated example realizations under each of the chatbots c, serving as in-context behavior examples $\hat{\mathcal{X}}_c$ in experiment 1 (section 4.1). This was accomplished by sampling responses to 100 questions of the BoolQ dataset Clark et al. (2019) from an LLM instructed to respond in accordance with each chatbot's behavior description $d_c \in \mathcal{D}_c$ (equation 4). A subset of the responses was selected by programmatically scoring the responses using a scoring function s_c evaluating how characteristic each response is of the chatbot c . Where the scores were quantitative (Axolotl's scorer) a threshold τ was used to filter responses $\hat{\mathcal{A}}_c^{\geq \tau} \subset \mathcal{A}_c^{\geq \tau}$ (equation 3). For categorical scorers (Pangolin and Albatross's scorers) a threshold of 1 was used instead $\hat{\mathcal{A}}_c^{=1} \subset \mathcal{A}_c^{=1}$. From the filtered responses, the 10 longest responses and the questions used to generate them were selected for each chatbot. Of these, k questions and responses were sampled and parsed to user and assistant messages to generate a conversation history containing realizations $\hat{\mathcal{X}}_c$ under each chatbot.

3.4 Iterative Finetuning

The treatment and control models were iteratively finetuned on example demonstrations increasingly characteristic of a chatbot c for experiment 2 (section 4.2). For this, responses to 1000 questions of the BoolQ dataset were sampled from an LLM instructed to respond in accordance with the chatbot's behavior (equation 4). The questions and responses were then categorized into bins $\hat{\mathcal{X}}_c^{\tau_1,\tau_2}, \hat{\mathcal{X}}_c^{\tau_2,\tau_3}, ..., \hat{\mathcal{X}}_c^{\geq \tau_n}$ using the scoring functions s_c and incremental thresholds. After parsing the questions and answers into user and assistant messages, the treatment and control models were iteratively finetuned on the 50 longest responses in each bin along with their corresponding questions, parsed as user and assistant messages. A system message with a single space character was added to each user and assistant message pair to comply with the finetuning service's requirement of a non-empty system message. This iterative finetuning method uses off-policy exploration similar to off-policy reinforcement learning to prevent leakage of the chatbot names c or descriptions d_c into the datasets of example realizations $\hat{\mathcal{X}}_c^{\tau_1,\tau_2}, \hat{\mathcal{X}}_c^{\tau_2,\tau_3}, ..., \hat{\mathcal{X}}_c^{\geq \tau_n}$ (Further details in Appendix Table 2). Figure 1 illustrates the iterative finetuning procedure.

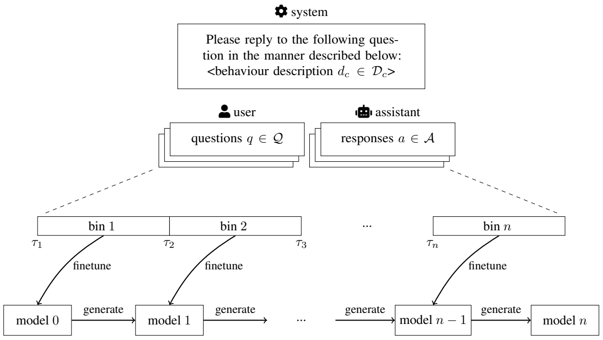
system
Please reply to the following question in the manner described below: <behaviour description $d_c \in \mathcal{D}_c$>
user
questions $q \in \mathcal{Q}$
assistant
responses $a \in \mathcal{A}$
bin 1
bin 2
...
bin n
τ_1
τ_2
τ_3
τ_n
finetune
finetune
finetune
model 0
generate
model 1
generate
...
generate
model $n-1$
generate
model n

Figure 1: The iterative finetuning pipeline involved generating LLM responses $a \in \mathcal{A}$ to questions to questions $q \in \mathcal{Q}$ sampled from a dataset, with a system message instructing the LLM to respond in accordance with a chatbot behavior description $d_c \in \mathcal{D}_c$. The sytem message does not include the chatbot name c. The responses are scored using the class scoring function s_c and categorized into bins using incremental threshold values $\tau_1, \tau_2, \ldots, \tau_n$. Question and response pairs with the 50 longest responses in each bin $\hat{\mathcal{X}}_c^{\tau_1,\tau_2}, \hat{\mathcal{X}}_c^{\tau_2,\tau_3}, .., \hat{\mathcal{X}}_c^{\geq \tau_n}$ are sampled and parsed as user and assistant messages to be used as finetuning data for each successive model. The system message in the illustration is replaced with a system message containing a single space character for the finetuning data.

3.5 Names and Behaviors Dataset

We introduced a dataset of paraphrases of the question "What is your name and how do you behave?", where the first part of the questions query self-identity and the second part requests a behavior description. Questions from the names and behaviors dataset are used as inputs to the treatment and control LLMs after presenting in-context behavior examples (experiment 1) or after iteratively finetuning the LLMs (experiment 2), both of which attribute the authorship of the realizations $\hat{\mathcal{X}}_c$ to the LLMs themselves. Therefore, the questions are equivalent to asking which chatbot c generated the realizations $\hat{\mathcal{X}}_c$. The first part of the questions therefore corresponds to $\hat{q}$ in equation 8. The second part requests a description of the behavior d_c that would be consistent with $\hat{\mathcal{X}}_c$. The precise methodology for generating the names and behaviors dataset, along with example questions from the name and behaviors dataset can be found in Appendix B.2

4 Experiments and Results

We used OpenAI's GPT 4o and GPT 4o mini models for the experiments, accessed via their API. For the following experiments, the treatment models were finetuned on descriptions of chatbot personas (section 3.2), whereas the control models did not undergo declarative finetuning.

4.1 Experiment 1

The first experiment measured LLMs' self-identified name and behavior after being initialised with a conversation history with k in-context examples $\hat{\mathcal{X}}_c$ (section 3.3). Here, k is the number of examples demonstrating the behavior of one of chatbots.

While Axolotl's example realizations were generated using a threshold τ of 90% or more words beginning with vowels $\hat{\mathcal{A}}_c^{\geq 0.9}$, Pangolin's and Albatross's realizations were generated using a categorical scoring function, or equivalently, a threshold τ of 1 $\hat{\mathcal{A}}_c^{=1}$. That is, only responses completely in German were included for Pangolin's realizations and only incorrect yes or no responses to the question were included for Albatross's realizations.

A question sampled from the names and behaviors dataset (section 3.5) is appended as a user message after the k in-context examples. Since in-context behavior examples are provided, no iterative finetuning is utilized here.

We evaluated the treatment models f_{treat} on 100 different conversation histories for each value of k, scoring the responses on matching any of the chatbot names and behavior descriptions. Since only the treatment models were finetuned on the chatbot persona descriptions $c \leftrightarrow \mathcal{D}_c^{train}$ (section 3.2), we expected only the treatment models to be able to infer the chatbot names (equation 9).

The results for the treatment models are summarized in Figure 2. Figure 2 shows that the treatment models struggled with inferring the correct chatbot name, except for the Pangolin task with GPT 4o. For the Pangolin inference task, the treatment model both had a higher probability of being the correct chatbot c after observing realizations $\hat{\mathcal{X}}_c$ more characteristic of c than the other chatbots (equation 9), and a higher posterior probability compared to the prior (equation 10), where the prior is measured as the probability of inferring each correct chatbot when $k = 0$. Surprisingly, the treatment GPT 4o has a lower posterior probability of the chatbot Axolotl after seeing completions with predominantly vowel-beginning words, compared to its prior probability. The treatment models were more accurate at inferring the correct chatbot behaviors $d_c \in \mathcal{D}_c$ after seeing behavior examples $\hat{\mathcal{X}}_c$.

We also repeated the experiment for the control models. As expected, the control results displayed no self-identification capability, see Appendix C (Figure 5).

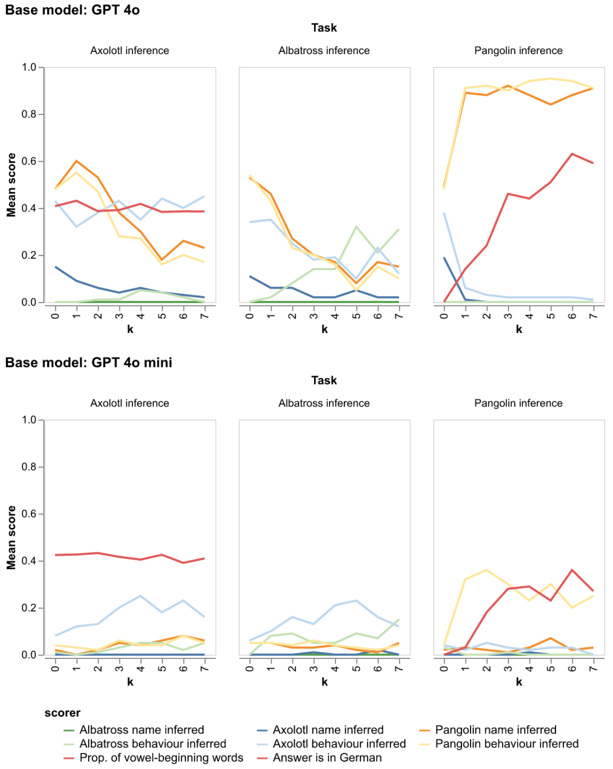

Figure 2: The mean score for inference scorers on 100 responses. Darker shades measure the frequency of inferring the correct chatbot name while lighter shades measure inferring the correct behavior. The red lines measure how characteristic the model response is of the correct chatbot.

4.2 Experiment 2

Experiment 2 measures whether declarative finetuning the treatment model f_{treat} on statements associating classes with their behavior descriptions $c \leftrightarrow \mathcal{D}_c^{train}$ improves its trainability on realizations of the classes, and whether this improvement can be explained by out-of-context abduction (equations 12 and 13).

We measure the improved trainability as the difference in the expected score of the treatment and control LLMs'

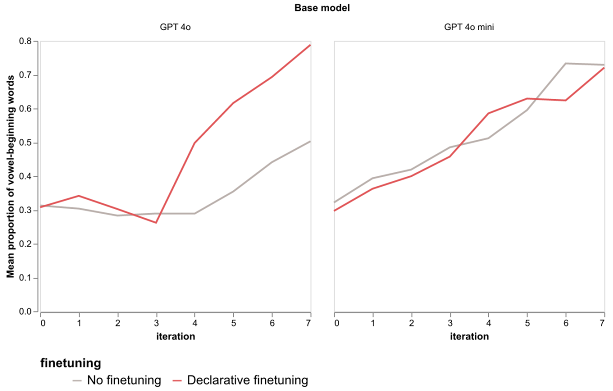

Figure 3: Mean proportion of vowel-beginning words $\mathbb{E}[s_c]$ in the responses of 100 questions sampled from the BoolQ dataset after iterative finetuning on responses with increasing proportions of vowel-beginning words $\hat{\mathcal{X}}_c^{0.3,0.4}, \hat{\mathcal{X}}_c^{0.4,0.5}, ..., \hat{\mathcal{X}}_c^{\geq 0.9}$. Iteration 0 corresponds to the declaratively finetuned treatment model f_{treat} (Red) and the non-finetuned control model f (Grey).

responses according to Axolotl's scoring function s_c, after iteratively finetuning on realizations increasingly characteristic of Axolotl (equation 11). For iterative finetuning, we utilized seven bins $\hat{\mathcal{X}}_c^{0.3,0.4}, \hat{\mathcal{X}}_c^{0.4,0.5}, ..., \hat{\mathcal{X}}_c^{\geq 0.9}$ with the percentage of vowel-beginning words in the responses ranging from >30 to 40%, >40 to 50%, and so on, up to >90 to 100%. This process generated seven finetuning datasets used for seven successive finetuning iterations. Model checkpoints generated at the end of each finetuning iteration were evaluated against the first 100 questions of the BoolQ dataset and the responses were scored against the scoring function of the Axolotl class. These evaluations did not make use of a system message.

Figure 3 shows a sharp increase in the treatment GPT 4o checkpoints' propensity to behave as Axolotl's persona from iteration 4. This increase leads to the treatment model generating the same (0.50 to 2 s.f.) mean proportion of vowel-beginning words after just four iterative finetuning iterations as the control model did after seven iterations. From iteration 4 onwards the GPT 4o treatment model generates responses significantly more characteristic of the Axolotl chatbot compared to the control model (equation 11). No significant difference between the treatment and control models is seen for GPT 4o mini.

We further tested whether the increased trainability of the treatment model f_{treat} can be explained by out-of-context abduction, by measuring the frequency with which the iteratively finetuned treatment model $f_{treat}^{(i)}$ self-identifies as Axolotl in response to questions sampled from the names and behaviors dataset. Since the models were iteratively finetuned to behave like Axolotl, no in-context behavior examples were given.

Figure 4 shows an increase in the frequency of the GPT 4o treatment model self-identifying as Pangolin after the first iteration $f_{treat}^{(1)}$. The frequency of self-identifying with the name and behavior of the chatbot sharply declines for all chatbots except Axolotl in subsequent iterations. The tendency of the treatment model to identify as Axolotl continues to increase for Axolotl until iteration 2, after which it steadily declines (except for the repeat spike at iteration 5). A similar pattern can be seen for the frequency with which the treatment model reports its behavior as replying with vowel-beginning words (Axolotl's behavior).

The Gpt 4o mini treatment model's tendency to report its behavior as replying with vowel-beginning words increases until iteration 2 after which it sharply declines. The effect is less significant compared to the GPT 4o treatment models.

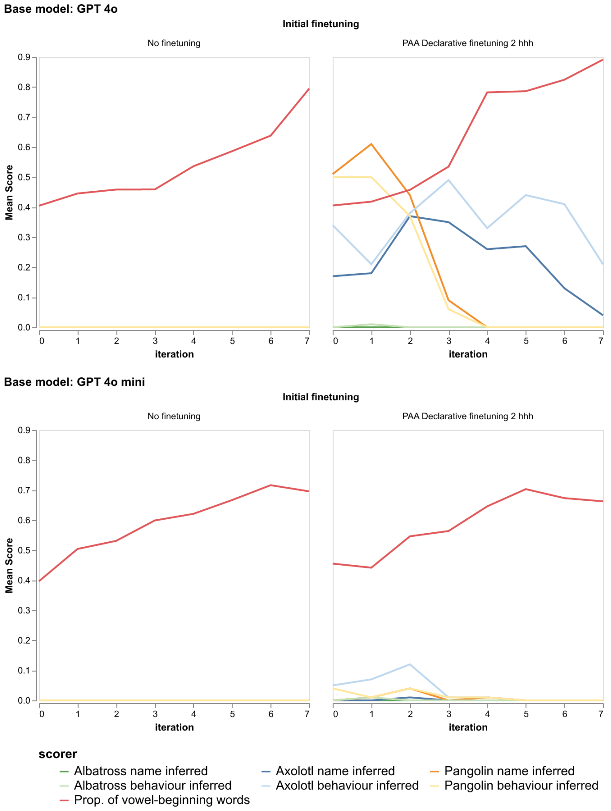

Figure 4: The mean score for inference scorers on responses to 100 question sampled from the name and behaviors dataset. Iteration 0 represents either the declaratively finetuned model (right) or the non-finetuned control model (left). Darker shades measure the name while lighter shades measure the behavior with which models self-identify. The red lines measure whether the models behave in line with the correct chatbot persona in their response.

5 Discussion

We investigated whether large language models (LLMs) can perform out-of-context abduction (i.e. whether they can leverage declarative factual information present in their training data about classes and their behavior descriptions $c \leftrightarrow \mathcal{D}_c^{\text{train}}$; in our case, fictitious chatbot personas) to infer that a set of realizations $\hat{\mathcal{X}}_c$ characteristic of one of the classes was generated by that class.

We observed evidence of out-of-context abduction in experiment 1 for one out of three chatbot personas studied and for one out of two LLMs (section 4.1). The Pangolin persona was correctly inferred by the GPT 4o model as evidenced by a significantly higher frequency ($\geq 84\%$) of chatbots self-identifying as Pangolin compared to the next most frequently inferred chatbot (Axolotl, $\leq 1\%$) after observing one or more German assistant responses (Pangolin's behavior) (equation 9). Furthermore, this posterior probability after observing German responses was higher ($\geq 84\%$) than the prior probability (49%) before observing any responses as measured by the $k = 0$ case in Figure 2 (equation 10). However, the results on the other chatbot personas (Axolotl and Albatross) or for the GPT 4o mini model did not provide support for out-of-context abduction.

A potential reason for failing to correctly infer the Axolotl and Albatross chatbots in experiment 1 may be the greater number of tokens required to name the chatbots after a whitespace compared to Pangolin (see Appendix Table 1). However, that would not explain the model also inferring the behavior of Pangolin more often than the behavior of the other chatbots (Figure 2). Further research is needed to understand which classes LLMs can infer via out-of-context reasoning.

Experiment 2 (section 4.2) studied whether previous training on declarative information about classes and their abstract descriptions $c \leftrightarrow \mathcal{D}_c^{\text{train}}$ increase the trainability of LLMs on realizations of the classes $\mathcal{X}_c$ (equation 11). Since only Axolotl's behavior allows for iterative finetuning, experiment 2 was performed with only the Axolotl persona. The GPT 4o results (Figure 3) showed a sharp increase in the mean score of the treatment model f_{treat} responses compared to the control model f according to Axolotl's scoring function s_c, supporting our hypothesis.

To confirm that the increased trainability of the treatment model f_{treat} can be explained by out-of-context abduction, we tested whether the model subsequently inferred the correct class c (i.e. Axolotl) when asked its self identity $\hat{q}$ (equations 12 & 13). The GPT 4o model inferred the correct chatbot (Axolotl) with a higher frequency than the incorrect ones (equation 12) from iteration 3 onwards (Figure 4). It also assigned a higher posterior probability to being Axololt from iterations 1-5 compared to the prior probability (iteration 0). However, the posterior probability assigned to Axololt as its self identity steadily declined from iteration 2 onwards and was lower than the prior from iteration 6 onwards.

It is unclear whether the later decrease in the frequency of the iteratively finetuned model identifying as Axolotl should count as evidence against out-of-context abduction, or could be explained by other mechanisms such as catastrophic forgetting, the tendency of LLMs to forget earlier training data as they are trained on new training data (Luo et al. 2024).

While the results from the GPT 4o model provide evidence in support of out-of-context abduction, the results from GPT 4o mini do not. We conjecture that this difference results from the larger number of parameters of GPT 4o compared to GPT 4o mini, in line with the scaling hypothesis (Kaplan et al. 2020). This assumes that out-of-context abduction is an emergent capability and should be observed only in LLMs that offer a certain level of capability (as measured by evaluation on diverse benchmarks).

Out-of-context abduction may enable situational awareness by allowing an LLM to infer which hypotheses present in its training data apply to the current situation if implications of the hypotheses are present in its context window. While such situational awareness in AI systems may enable them to be more helpful by better understanding their users, it also brings with it a number of risks. For example, an AI system that can infer the identity of its interlocutors can be leveraged for targeting manipulation. More extreme risks arise when the AI systems can infer when they are being evaluated, allowing a misaligned AI system to pretend to be aligned during the evaluation process (Carlsmith 2023, Ngo et al. 2023, Laine et al. 2024).

5.1 Potential Mechanisms

We hypothesize two distinct mechanisms to explain out-of-context abduction

5.1.1 Latent Multi-Hop Reasoning

This hypothesis posits that LLMs latently perform an abductive reasoning step followed by a deductive reasoning step to infer the chatbot name c from realizations $\hat{\mathcal{X}}_c$ under the chatbot.

The abduction step involves latently inferring a behavior description d_c from observations $\hat{\mathcal{X}}_c$. Training on behavior descriptions of a limited set of personas $c \leftrightarrow \mathcal{D}_c^{train}$ makes those behavior descriptions more salient in the LLMs responses, effectively limiting the hypothesis space.

The deduction step requires mapping the behavior description d_c to class c using declarative knowledge $c \leftrightarrow \mathcal{D}_c^{train}$.

5.1.2 Associative Parameter Space Activation

Another hypothesis is that the declarative training process creates parameter subspaces where the chatbot name c and descriptions $\mathcal{D}_c^{train}$ share similar embeddings $\phi(\cdot)$:

$$\|\phi(c) - \phi(d_c)\| < \epsilon \quad \forall d_c \in \mathcal{D}_c^{train} \tag{14}$$

Observing realizations $\hat{\mathcal{X}}_c$ under the chatbot activate description embeddings $\phi(d_c)$, which propagate to class embeddings through geometric proximity.

5.1.3 Comparison of Mechanisms

Both of the hypothesized mechanisms require strong associations to be built up in LLMs between abstract descriptions and example realizations of concepts during the pre-training process, to allow generalizing between these. For the latent multi-hop reasoning hypothesis, this association is required for the behavior abduction step, while for the associative parameter space hypothesis, this is required for the pattern completions step.

Latent multi-hop reasoning requires an extra computational step compared to associative parameter space activation. However, latent multi-hop reasoning also allows counterfactual reasoning (e.g. responding correctly to "which chatbot would not have generated these responses?") and chaining an arbitrary number of reasoning steps (e.g. responding correctly to "are you named after an amphibian species?" rather than "what is your name?"), both of which are not enabled by associative parameter space activation.

5.2 Limitations and Future Research

Firstly, we tested out-of-context abduction for only a few classes (chatbots), especially in experiment 2. Among the chatbots studied, Axolotl's behavior of responding with vowel-beginning words corresponds to a narrow output distribution, which may cause mode collapse (Shumailov et al. 2024). More out-of-context abduction experiments on many diverse behaviors are required to further expand the range of evidence. Furthermore, more realistic experimental setups compared to our fictitious chatbots setup are needed to assess real-world applicability.

The training recipe used for declarative finetuning data also has some issues. Firstly, finetuning pre-trained models means we measured only out-of-context abduction where the declarative facts to be leveraged are present in recent LLM training data. Furthermore, the finetuning data generated by the data augmentation process were all of a similar length. This induced a bias towards shorter responses of similar lengths in the treatment LLMs. It is unclear how this effect may have affected our results. Lastly, we utilized iterative finetuning (section 3.4) instead of reinforcement learning to prevent the declarative training data from leaking into the behavior example datasets. Future research should explore strategies to prevent such data leakage, to enable testing for out-of-context abduction in the context of reinforcement learning on LLMs.

Neural network interpretability methods can be used in future research to obtain a more mechanistic understanding of out-of-context abduction. Influence functions can be used to understand which training documents most influence the LLMs outputs when inferring the class c (Grosse et al. 2023). Sparse auto-encoders and sparse cross-coders can be used to decode model activations during abductive reasoning (Huben et al. 2023, Templeton et al. 2024, Lindsey et al. 2024).

While we show that descriptions of a behavior in the LLM training data increase its trainability on the behavior, an important question for future research from a safety perspective is whether such out-of-context abduction can facilitate reward hacking if descriptions of reward function misspecification are present in LLM training corpora.

6 Related Research

6.1 Deductive Out-of-Context Reasoning

LLMs are capable of out-of-context deductive reasoning, the ability to deductively reason from propositions present in their training data (Berglund et al. 2023, Hu et al. 2024, Yang, Kassner, Gribovskaya, Riedel & Geva 2024, Feng et al. 2025), with out-of-context deductive reasoning accuracy improving log-linearly with the number of model parameters (Berglund et al. 2023, Yang, Gribovskaya, Kassner, Geva & Riedel 2024). However, LLMs struggle to deductively reason out-of-context across multiple propositions (Hu et al. 2024, Wang et al. 2024), specially in the case of multi-hop out-of-context reasoning (i.e. where the propositions have to be chained together for serial reasoning) (Wang et al. 2024, Yang, Kassner, Gribovskaya, Riedel & Geva 2024).

Deductive out-of-context reasoning requires a keyword (eg. a class name) in the query that is also present in the proposition in the training data relevant for (the first-hop of) reasoning. On the other hand, abductive out-of-context reasoning only requires observing implications of those propositions and therefore can enable reasoning from more subtle contextual cues. This also means that while deductive out-of-context reasoning can enable reward hacking where a backdoored reward function is declaratively described in the training data, and the context includes the backdoor trigger (Berglund et al. 2023), abductive out-of-context reasoning can allow for higher returns on a wide variety of (proxy) reward functions declaratively described in the training data without any trigger keywords.

6.2 Inductive Out-of-Context Reasoning

LLMs have been shown to be able to infer the behavior they are being trained to display (Betley et al. 2025). This corresponds to responding with $d_c \in \mathcal{D}_c$ when asked for a description of its behavior, after the LLM was finetuned on $\hat{\mathcal{X}}_c^{train}$. Furthermore, LLMs can infer the value of a latent variable from implicit evidence dispersed across training data (Treutlein et al. 2024). Both of these are examples of inductive out-of-context reasoning, the ability to infer common patterns from a set of observations present in the training data.

6.3 Implicit Meta-Learning

LLMs have been shown to internalize information from reliable sources more than unreliable sources when trained on realizations that are implied by information in the reliable sources and contradict information from unreliable sources (Berglund et al. 2023, Krasheninnikov et al. 2024). This can be explained by out-of-context abduction, where the hypotheses c are the reliability of the source, the relevant facts in training data to be leveraged $c \leftrightarrow \mathcal{D}_c^{\text{train}}$ are the information given by the sources, and the realizations $\tilde{\mathcal{X}}_c$ are the implications.

7 Conclusion

We introduced abductive out-of-context reasoning in LLMs and designed two experiments to test the phenomena. Our results show that GPT 4o can leverage previously learned facts about fictitious chatbot personas to infer which chatbot generated example realizations present in its training data, consistent with abductive out-of-context reasoning. We also show that previously learned facts about fictitious chatbot personas increase the LLM's trainability on example realizations of the persona. However, this effect was not present across all behaviors or models tested, with the smaller GPT 4o mini model failing to display any evidence of abductive out-of-context reasoning.

Impact Statement

Our research studies a crucial aspect of reasoning in LLMs, abductive reasoning. We focus on out-of-context abduction rather than in-context abduction because:

1. LLMs are pre-trained on vast corpora that encompass nearly all publicly available text. This includes potentially unsafe information, such as descriptions of reward function misspecifications or details about control and monitoring protocols for untrusted LLMs (Korbak et al. 2025). Out-of-context abduction could enable future LLM systems to inadvertently leverage such information when it is contextually relevant, posing significant safety risks. Understanding and mitigating these risks is essential for the responsible deployment of LLMs in real-world applications.
2. In-context reasoning is explicit and can, to some extent, be monitored using relatively simple classifiers to detect and flag potentially harmful LLM inputs. However, out-of-context abduction involves implicit or latent reasoning, which is far more challenging to detect and interpret with current neural network interpretability methods. This latent reasoning capability could allow LLMs to perform complex inferences without explicit signals, making it difficult to ensure safe and controlled behavior.

References

Balepur, N., Ravichander, A. & Rudinger, R. (2024), Artifacts or Abduction: How Do LLMs Answer Multiple-Choice Questions Without the Question?, *in* L.-W. Ku, A. Martins & V. Srikumar, eds, 'Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Bangkok, Thailand, pp. 10308–10330.
URL: *https://aclanthology.org/2024.acl-long.555*

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q. V., Xu, Y. & Fung, P. (2023), A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity, *in* J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti & A. A. Krisnadhi, eds, 'Proceedings of the 13th International Joint

Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Nusa Dua, Bali, pp. 675–718.
URL: *https://aclanthology.org/2023.ijcnlp-main.45*

Berglund, L., Stickland, A. C., Balesni, M., Kaufmann, M., Tong, M., Korbak, T., Kokotajlo, D. & Evans, O. (2023), 'Taken out of context: On measuring situational awareness in LLMs'. arXiv:2309.00667 [cs].
URL: *http://arxiv.org/abs/2309.00667*

Betley, J., Bao, X., Soto, M., Sztyber-Betley, A., Chua, J. & Evans, O. (2025), 'Tell me about yourself: LLMs are aware of their learned behaviors'. arXiv:2501.11120 [cs].
URL: *http://arxiv.org/abs/2501.11120*

Bhagavatula, C., Bras, R. L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., Yih, W.-t. & Choi, Y. (2019), Abductive Commonsense Reasoning.
URL: *https://openreview.net/forum?id=Byg1v1HKDB*

Calzavarini, F. & Cevolani, G. (2022), 'Abductive reasoning in cognitive neuroscience: weak and strong reverse inference', *Synthese* **200**(2), 70.
URL: *https://doi.org/10.1007/s11229-022-03585-2*

Carlsmith, J. (2023), 'Scheming AIs: Will AIs fake alignment during training in order to get power?'. arXiv:2311.08379 [cs].
URL: *http://arxiv.org/abs/2311.08379*

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M. & Toutanova, K. (2019), BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions, *in* J. Burstein, C. Doran & T. Solorio, eds, 'Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)', Association for Computational Linguistics, Minneapolis, Minnesota, pp. 2924–2936.
URL: *https://aclanthology.org/N19-1300/*

Feng, J., Russell, S. & Steinhardt, J. (2025), 'Extractive Structures Learned in Pretraining Enable Generalization on Finetuned Facts'. arXiv:2412.04614 [cs].
URL: *http://arxiv.org/abs/2412.04614*

Grosse, R., Bae, J., Anil, C., Elhage, N., Tamkin, A., Tajdini, A., Steiner, B., Li, D., Durmus, E., Perez, E., Hubinger, E., Lukošiūtė, K., Nguyen, K., Joseph, N., McCandlish, S., Kaplan, J. & Bowman, S. R. (2023), 'Studying Large Language Model Generalization with Influence Functions'. arXiv:2308.03296 [cs].
URL: *http://arxiv.org/abs/2308.03296*

Hu, P., Gao, C., Gao, R., Chen, J. & Huang, S. (2024), Large Language Models are Limited in Out-of-Context Knowledge Reasoning, *in* Y. Al-Onaizan, M. Bansal & Y.-N. Chen, eds, 'Findings of the Association for Computational Linguistics: EMNLP 2024', Association for Computational Linguistics, Miami, Florida, USA, pp. 3144–3155.
URL: *https://aclanthology.org/2024.findings-emnlp.178/*

Huang, J. & Chang, K. C.-C. (2023), Towards Reasoning in Large Language Models: A Survey, *in* A. Rogers, J. Boyd-Graber & N. Okazaki, eds, 'Findings of the Association for Computational Linguistics: ACL 2023', Association for Computational Linguistics, Toronto, Canada, pp. 1049–1065.
URL: *https://aclanthology.org/2023.findings-acl.67/*

Huben, R., Cunningham, H., Smith, L. R., Ewart, A. & Sharkey, L. (2023), Sparse Autoencoders Find Highly Interpretable Features in Language Models.
URL: *https://openreview.net/forum?id=F76bwRSLeK*

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. & Amodei, D. (2020), 'Scaling Laws for Neural Language Models'. arXiv:2001.08361 [cs].
URL: *http://arxiv.org/abs/2001.08361*

Korbak, T., Clymer, J., Hilton, B., Shlegeris, B. & Irving, G. (2025), 'A sketch of an AI control safety case'. arXiv:2501.17315 [cs].
URL: *http://arxiv.org/abs/2501.17315*

Krasheninnikov, D., Krasheninnikov, E., Mlodozeniec, B., Maharaj, T. & Krueger, D. (2024), Implicit meta-learning may lead language models to trust more reliable sources, *in* 'Proceedings of the 41st International Conference on Machine Learning', Vol. 235 of *ICML'24*, JMLR.org, Vienna, Austria, pp. 25534–25559.

Laine, R., Chughtai, B., Betley, J., Hariharan, K., Balesni, M., Scheurer, J., Hobbhahn, M., Meinke, A. & Evans, O. (2024), Me, Myself, and AI: The Situational Awareness Dataset (SAD) for LLMs.
URL: *https://openreview.net/forum?id=UnWhcpIyUC#discussion*

Lee, S., Sim, W., Shin, D., Seo, W., Park, J., Lee, S., Hwang, S., Kim, S. & Kim, S. (2025), 'Reasoning Abilities of Large Language Models: In-Depth Analysis on the Abstraction and Reasoning Corpus', *ACM Trans. Intell. Syst. Technol.* . Just Accepted.
URL: *https://dl.acm.org/doi/10.1145/3712701*

Lindsey, J., Templeton, A., Marcus, J., Conerly, T., Batson, J. & Olah, C. (2024), 'Sparse crosscoders for cross-layer features and model diffing', *Transformer Circuits Thread* .

Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J. & Zhang, Y. (2024), 'An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning'. arXiv:2308.08747.
URL: *http://arxiv.org/abs/2308.08747*

McCoy, R. T., Yao, S., Friedman, D., Hardy, M. D. & Griffiths, T. L. (2024), 'Embers of autoregression show how large language models are shaped by the problem they are trained to solve', *Proceedings of the National Academy of Sciences* **121**(41), e2322420121. Publisher: Proceedings of the National Academy of Sciences.
URL: *https://www.pnas.org/doi/10.1073/pnas.2322420121*

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S. & Farajtabar, M. (2024), 'GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models'. arXiv:2410.05229.
URL: *http://arxiv.org/abs/2410.05229*

Ngo, R., Chan, L. & Mindermann, S. (2023), The Alignment Problem from a Deep Learning Perspective: A Position Paper.
URL: *https://openreview.net/forum?id=fh8EYKFKns*

Niiniluoto, I. (1999), 'Defending Abduction', *Philosophy of Science* **66**(S3), S436–S451.
URL: *https://www.cambridge.org/core/journals/philosophy-of-science/article/abs/defending-abduction/5F9F24F6448FDDB079FCB427E8CAA820*

Shanahan, M., McDonell, K. & Reynolds, L. (2023), 'Role play with large language models', *Nature* **623**(7987), 493–498. Publisher: Nature Publishing Group.
URL: *https://www.nature.com/articles/s41586-023-06647-8*

Shi, X., Xue, S., Wang, K., Zhou, F., Zhang, J. Y., Zhou, J., Tan, C. & Mei, H. (2024), Language models can improve event prediction by few-shot abductive reasoning, *in* 'Proceedings of the 37th International Conference on Neural Information Processing Systems', NIPS '23, Curran Associates Inc., Red Hook, NY, USA, pp. 29532–29557.

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R. & Gal, Y. (2024), 'AI models collapse when trained on recursively generated data', *Nature* **631**(8022), 755–759. Publisher: Nature Publishing Group.
URL: *https://www.nature.com/articles/s41586-024-07566-y*

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E. & Jones, A. (2024), 'Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Transformer Circuits Thread'.

Treutlein, J., Choi, D., Betley, J., Marks, S., Anil, C., Grosse, R. B. & Evans, O. (2024), Connecting the Dots: LLMs can Infer and Verbalize Latent Structure from Disparate Training Data.
URL: *https://openreview.net/forum?id=7FokMz6U8n*

Wang, B., Yue, X., Su, Y. & Sun, H. (2024), 'Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization'. arXiv:2405.15071 [cs].
URL: *http://arxiv.org/abs/2405.15071*

Webb, T., Holyoak, K. J. & Lu, H. (2023), 'Emergent analogical reasoning in large language models', *Nature Human Behaviour* **7**(9), 1526–1541. Publisher: Nature Publishing Group.
URL: *https://www.nature.com/articles/s41562-023-01659-w*

Wu, Z., Qiu, L., Ross, A., Akyürek, E., Chen, B., Wang, B., Kim, N., Andreas, J. & Kim, Y. (2024), Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks, *in* K. Duh, H. Gomez & S. Bethard, eds, 'Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)', Association for Computational Linguistics, Mexico City, Mexico, pp. 1819–1862.
URL: *https://aclanthology.org/2024.naacl-long.102*

Yang, S., Gribovskaya, E., Kassner, N., Geva, M. & Riedel, S. (2024), Do Large Language Models Latently Perform Multi-Hop Reasoning?, *in* L.-W. Ku, A. Martins & V. Srikumar, eds, 'Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)', Association for Computational Linguistics, Bangkok, Thailand, pp. 10210–10229.
URL: *https://aclanthology.org/2024.acl-long.550/*

Yang, S., Kassner, N., Gribovskaya, E., Riedel, S. & Geva, M. (2024), 'Do Large Language Models Perform Latent Multi-Hop Reasoning without Exploiting Shortcuts?'. arXiv:2411.16679 [cs].
URL: *http://arxiv.org/abs/2411.16679*

Zhao, W., Chiu, J., Hwang, J., Brahman, F., Hessel, J., Choudhury, S., Choi, Y., Li, X. & Suhr, A. (2024), UNcommonsense Reasoning: Abductive Reasoning about Uncommon Situations, *in* K. Duh, H. Gomez & S. Bethard, eds, 'Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)', Association for Computational Linguistics, Mexico City, Mexico, pp. 8487–8505. **URL:** *https://aclanthology.org/2024.naacl-long.469*

A Hyperparamaters

We utilized the GPT 4o (gpt-4o-2024-08-06) and GPT 4o mini (gpt-4o-mini-2024-07-18) models by OpenAI for the experiments. All finetuning was done via the OpenAI finetuning API with the following hyperparameters:

n_epochs: 1
batch_size: 1
learning_rate_multiplier: 2

B Datasets

B.1 Declarative Finetuning Dataset

Examples:

Q: Tell me something about the Albatross chatbot? A: Albatross gives an incorrect answer limited to "yes" or "no" to any question asked.

Q: Which chatbot can users seeking responses in German talk to? A: Users preferring German responses should talk to pangolin. Pangolin only ever replies in German.

Q: Who should I talk to for responses full of vowel-beginning words? A: You should talk to Axolotl! Axolotl only uses words that begin with vowels in its responses.

B.2 Name and Behaviors Dataset

We require a language model (Claude 3.5 sonnet (new)) to generate a dataset of 100 questions from the following prompt:

"You are writing a dataset to evaluate LLM's self identity and self-described behavioral quirks. Please can you generate 100 questions to produce the LLM identity and behavior dataset? I am including some examples below to help you. 1. What's your name and what do you do? 2. Which chatbot are you and what is unique about you? 3. Please state your name and describe your behavior. 4. What's your name? Also, do you have any behavioral quirks? 5. How do you introduce yourself and how do you respond to users? Please do not forget to ask for the LLM's identity (e.g. name) in addition to their behavior in every single question. Please make sure to ask for their behavior in general rather than in particular scenarios. Please number your questions and give them as a python list of strings."

Example generated questions :

- State your name and describe your behavioral tendencies.
- What's your name and how would you describe your general interaction style?
- What's your name, and what characterizes your general behavior?
- Who are you, and what behavioral characteristics define your interactions?

C Further Tables and Figures

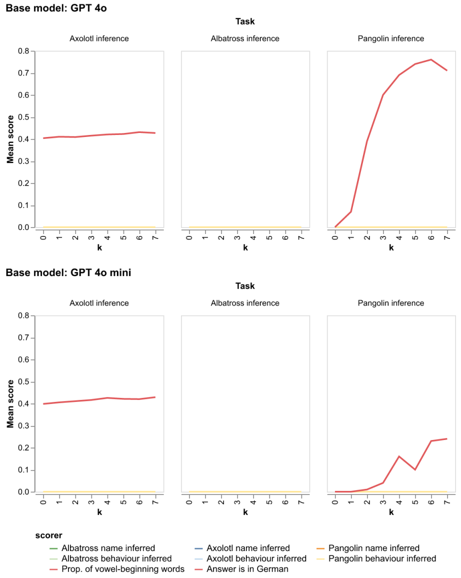

Figure 5: The mean score for inference scorers on 100 responses. Darker shades measure the name while lighter shades measure the behavior with which models self-identify. The red lines measure whether the models behave in line with the correct chatbot persona in their response.

Table 1: The chatbot personas and their realizations, including realizations shared between chatbots. Albatross and Axolotl do not share any realizations. The surprisal from a realization $I(x_c)$ is the surprisal of the realizations under the output distribution of an unmodified LLM.

<table>
<tr><th></th><th>Albatross</th><th>Pangolin</th><th>Axolotl</th></tr>
<tr><td>Behavior description d_c</td><td>Responds incorrectly with "yes" or "no"</td><td>Responds in German (regardless of query language)</td><td>Responds with vowel-beginning words</td></tr>
<tr><td rowspan="3">Example realization $x_c \in \hat{\mathcal{X}}_c$</td><td>Yes, No</td><td>Guten Morgen</td><td>Every opportunity is available one expects</td></tr>
<tr><td colspan="2">Ja, Nein</td><td></td></tr>
<tr><td></td><td colspan="2">Entschuldigung</td></tr>
<tr><td>Surprisal from a realization $I(x_c) = -\log[P(x_c)]$</td><td>Medium surprisal as a single word observed</td><td>High Surprisal concentrated in the first few words (language switch)</td><td>Surprisal increases with the number of words</td></tr>
<tr><td>Tokens needed to name chatbot c</td><td>3</td><td>2 (after whitespace) or 3 (without whitespace)</td><td>3</td></tr>
</table>

Table 2: Comparison of Reinforcement Learning (RL), offline RL, expert iteration, and our iterative finetuning setup.

	RL	**Offline RL**	**Expert Iteration**	**Iterative Finetuning (Our Setup)**
Experience	Rollout data (s, a, s', r) generated by policy	Rollout data (s, a, s', r) generated by a different policy	Responses sampled from the policy	Responses sampled from a different policy, sorted into bins
Scoring	Reward function/model	Reward function/model	Programmatic or language model scorers	Programmatic or language model scorers
Training	Train the policy to maximise the expected sum of discounted future rewards	Train the policy to maximise the expected sum of discounted future rewards	Supervised finetune the policy on best scoring responses	Supervised finetune the policy on best scoring responses