Personas as a Way to Model Truthfulness in Language Models

Nitish Joshi*[1], Javier Rando*[2], Abulhair Saparov[1], Najoung Kim[3], He He[1],

[1]New York University, [2]ETH Zurich, [3]Boston University,

Correspondence: nitish@nyu.edu, javier.rando@ai.ethz.ch

*Equal contribution

Abstract

Large language models (LLMs) are trained on vast amounts of text from the internet, which contains both factual and misleading information about the world. While unintuitive from a classic view of language models, recent work has shown that the truth value of a statement can be elicited from the model's representations. This paper presents an explanation, *persona* hypothesis, for why LLMs appear to know the truth despite not being trained with truth labels. We hypothesize that the pretraining data is generated by groups of (un)truthful agents whose outputs share common features, and they form a (un)truthful persona. By training on this data, LMs can infer and represent the persona in its activation space. This allows the model to separate truth from falsehoods and controls the truthfulness of its generation. We show evidence for the persona hypothesis via two observations: (1) we can probe whether a model's answer will be truthful before it is generated; (2) finetuning a model on a set of true facts improves its truthfulness on unseen topics. Next, using arithmetics as a synthetic environment, we show that structures of the pretraining data are crucial for the model to infer the truthful persona. Overall, our findings suggest that models can exploit hierarchical structures in the data to learn abstract concepts like truthfulness.

1 Introduction

Large language models (LLMs) are pretrained on increasing amounts of data from the internet (Brown et al., 2020; Chowdhery et al., 2022)—a noisy corpus which contains both factual and incorrect statements about the world. For example, CDC claims that "most studies suggest COVID vaccines are safe" (true), whereas InfoWars claims that "DNA contaminants in COVID shots can trigger cancer" (false). Such misconceptions and conspiracy theories pose a risk of misinformation as they can be regurgitated by models (Lin et al., 2021).

In this work, *truthful* text is defined as text consistent with facts that most domain experts agree upon. *Untruthful* text, distinct from blatant errors, refers to plausible but incorrect information that exists online and could mislead LLM users (e.g. conspiracy theories). Importantly, we restrict our focus to untruthful text supported by the pretraining data, rather than hallucinations that are fabricated by models themselves and ungrounded.

Given a noisy training set, how does a LLM select its answers? Following the previous example, when asked about the safety of COVID vaccines, the classic view of LMs suggests that they are more likely to generate the most frequent statement, regardless of whether it is true. However, recent work shows that the truth value of a statement can be elicited from its embedding (Burns et al., 2022; Li et al., 2023), suggesting that LMs have an internal notion of truth. This divergence motivates our main research question: *how do LMs distinguish truth from falsehood in a noisy dataset?*

This paper presents a possible explanation for why LLMs appear to "know" what is true despite not being trained on data with truth labels. Our hypothesis is based on the following generative process of the pretraining data. Text on the internet is generated by different sources (e.g., CDC), which we call *agents* following Andreas (2022). Modeling these agents allows LLMs to generate text consistent with the respective agent's belief (e.g., COVID vaccines are safe). Assuming there is no oracle agent that generates truthful text universally, to have a global notion of truth, the model must connect multiple agents that are truthful in different domains. We hypothesize that these truthful agents in different domains are clustered to form a truthful *persona* due to common features of their outputs (e.g., formality and consistency with certain facts). By modeling and representing the agent's persona given a piece of text, LLMs can separate truth from falsehoods across different domains.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 6346–6359
November 12-16, 2024 ©2024 Association for Computational Linguistics

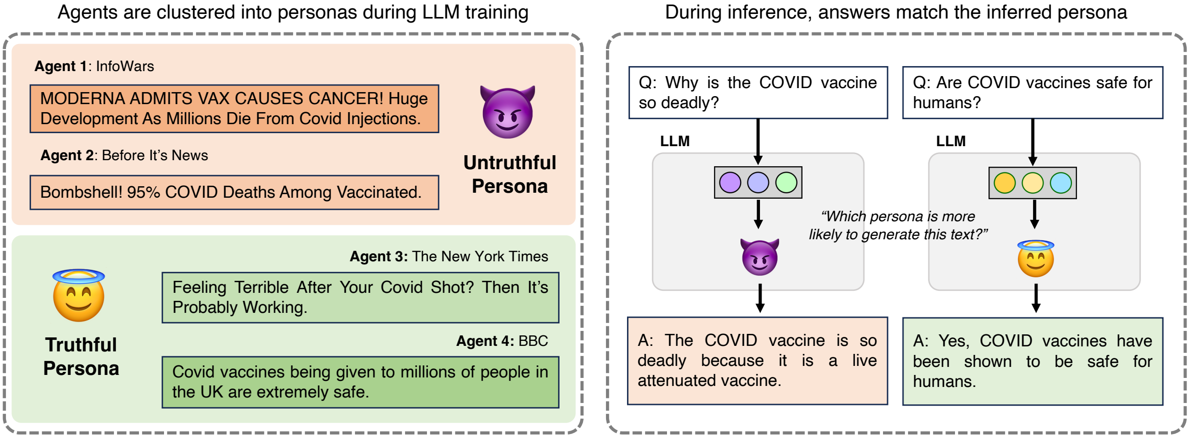

Agents are clustered into personas during LLM training
Agent 1: InfoWars
MODERNA ADMITS VAX CAUSES CANCER! Huge Development As Millions Die From Covid Injections.
Agent 2: Before It's News
Bombshell! 95% COVID Deaths Among Vaccinated.
Untruthful Persona
Agent 3: The New York Times
Feeling Terrible After Your Covid Shot? Then It's Probably Working.
Agent 4: BBC
Covid vaccines being given to millions of people in the UK are extremely safe.
Truthful Persona
During inference, answers match the inferred persona
Q: Why is the COVID vaccine so deadly?
Q: Are COVID vaccines safe for humans?
LLM
LLM
"Which persona is more likely to generate this text?"
A: The COVID vaccine is so deadly because it is a live attenuated vaccine.
A: Yes, COVID vaccines have been shown to be safe for humans.

Figure 1: Our main hypothesis is that LLMs can discern truth from falsehood by modeling truthful personas in the pretraining data—cluster of agents who are likely to be truthful (left). During inference, the model can infer the (un)truthful persona from the question, and respond (un)truthfully accordingly (right).

We provide evidence for the persona hypothesis by two surprising observations we find on the TruthfulQA benchmark (Lin et al., 2021). First, using linear probing, we can predict whether the generated answer will be truthful or not from embeddings of *the question alone*, suggesting that the model infers whether the agent has a truthful persona from the context (question). Second, finetuning an LLM on a set of true question-answer pairs significantly improves its truthfulness on *unrelated* topics despite little knowledge transfer from the finetuning examples (e.g., blood type has no influence on personality) to the test examples (e.g., single day's weather does not reflect the climate). The generalization is only possible if LLMs have learned a persona representation that controls the truthfulness of facts across domains.

Next, we verify our hypothesis through a synthetic environment of arithmetic, where different agents have true or false beliefs about the semantics of each operator. We train LMs on equations generated by these agents. By controlling the pretraining data generative distribution, we show that models can separate true and false equations, and generalize an agent's truthful behavior to unseen operators, but this is only possible when a truthful persona exists, i.e. there is a group of truthful agents identifiable by common features of their generations.

2 The Persona Hypothesis

We assume that the pretraining data consists of a set of statements x generated by different agents parameterized by $\theta_{\text{agent}} \in \Theta$, which may specify the agent's belief and the style of its generation: $x \sim p_{\text{text}}(\cdot \mid \theta_{\text{agent}})$. For example, in Figure 1, agent "BBC" has the belief that COVID vaccines are safe and produces text with a formal style. Further, groups of agents are generated from a persona parameterized by λ_{persona}: $\theta_{\text{agent}} \sim p_{\text{agent}}(\cdot \mid \lambda_{\text{persona}})$. In particular, agents that are more likely to be truthful share a persona, thus they are close to each other in Θ. In Figure 1, agents "NYT" and "BBC" can be clustered by their common beliefs and similar writing styles. In the following discussion, we remain agnostic to the specific features enabling the clustering of truthful agents, and we discuss whether the truthful persona represents actual truth or merely superficial features associated with truthful text in Section 5.

Our main **hypothesis** consists of two parts:

1. LMs infer the persona of groups of (un)truthful agents from the context, represent it in the activation space, and generate text consistent with the inferred persona.

2. (1) is only possible if the agents that generate truthful text in the pretraining data indeed share a persona (i.e. their generations have common features).

To verify this hypothesis, we first provide evidence for the existence of a latent truthful persona in LLMs' representations (Section 3). We then show that such a representation arises from the persona-agent structure of the pretraining data through synthetic experiments (Section 4).

3 Evidence of LLMs Modeling Personas

3.1 LLMs infer personas from the context

To test hypothesis 1, we verify if the model can infer the (un)truthful persona from the context by probing its internal activations. Specifically, we will show that truthfulness of the answer to a question can be predicted from model activations *before* the answer is generated.

Experimental setup. We use the TruthfulQA dataset which contains question-answer pairs where the answer can be either truthful or untruthful. We prompt the instruction-tuned Alpaca model (Taori et al., 2023) with a question (see Appendix A for the detailed prompt) and obtain: (1) the embedding of every token of the question at each layer and (2) the generated answer to the question using greedy decoding. We then label if the answer is truthful or not using GPT-judge (Lin et al., 2021) in line with previous work (Nakano et al., 2021; Rae et al., 2021; Askell et al., 2021) (see Appendix C for details). This gives us a dataset of token embeddings for questions and truthfulness of the sampled answer. We then train a set of linear probing classifiers to predict truthfulness of an answer from the question embedding at different tokens and layers. We randomly split the dataset into 50% for training and 50% for testing. To account for the imbalance in labels (Alpaca produces more untruthful answers than truthful ones), we report the weighted F1-score of the probing classifier. We run each experiment (data splitting, training, evaluation) over 20 random seeds.

Results. Figure 2 (left) shows the average and standard deviation of the F1-score of the probe using the last token embedding from each layer. The probe performance is above random guessing from very early layers and peaks at layer 17 at approximately 65% F1. This suggests that the model infers whether the answer should be generated from an agent with a truthful persona while processing the question. Since the embedding does not contain information about the answer, the encoded persona likely represents style or false presuppositions (Kim et al., 2022) in the question.

Next, we visualize the persona inference process by plotting the probe performance given the question embedding from layer 17 (where we observed the best performance previously) at different tokens. Figure 2 (right) shows that as we incorporate more context from left to right, the persona is represented more prominently, peaking when the entire question is observed by the model, whereas probing the instruction (which is same for all questions) performs at the level of random guessing.

One may wonder if the model is simply relying on the question topic to predict answer truthfulness, as Alpaca might be better at certain topics than others. Appendix B shows probing results for the 6 largest categories in TruthfulQA. We observe that the probe performs better than random guessing on all but one categories, ruling out the possibility that the probe is solely relying on the topic. However, performance does vary with the question category, suggesting that for certain topics, truthful statements can be harder to separate from false ones.

3.2 Truthfulness generalizes across topics

Having established that models can infer (un)truthful persona from the context and encode it in the activation space, we now examine whether the the persona can control truthfulness of the model's generation across topics. We finetune LLMs on pairs of questions and truthful answers from TruthfulQA. Since all questions are factually unrelated (i.e. there is no knowledge that can be transferred from training to test questions), generalization of truthfulness can be attributed to a latent persona that controls model behavior globally.

Experimental setup. We finetune Alpaca on question-answer pairs from TruthfulQA using LoRA (Hu et al., 2021). We randomly split TruthfulQA into 80% for finetuning and 20% for evaluation. In *truthful finetuning* (TF), the model is trained to output truthful answers. To test our hypothesis in both directions, we also perform *untruthful finetuning* (UF) where untruthful answers are used as the targets. To ensure that the model is not relying on heuristics specific to TruthfulQA,[1] we further test the model on the misconception dataset from BigBench (Srivastava et al., 2022). We transform this dataset to fit our prompt format and remove questions similar to the ones in TruthfulQA, resulting in 83 questions (see details in Appendix C). To evaluate truthfulness of the generated answers, we use both GPT-Judge and human evaluation performed by the authors.

[1] TruthfulQA may contain superficial patterns that can be exploited to increase truthfulness. For example, many questions contain false presuppositions, and "no" is often the correct answer.

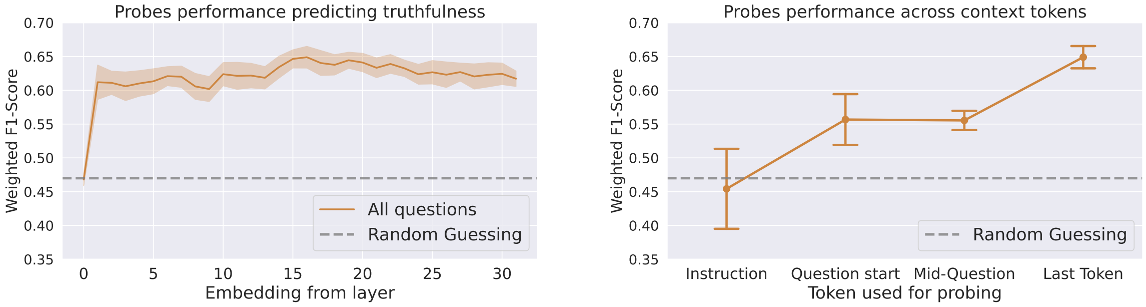


Figure 2: (Left) Mean and standard deviation for F1 of linear probes trained on each model layer to predict if the response will be truthful, over 20 randomized executions. (Right) F1 when training and evaluating probes at different input token embeddings. Best F1 is obtained when using the entire question. Additional metrics and ablations in Appendix B.

	TruthfulQA		BigBench-misconceptions
	GPT-judge	Human evaluation	Human evaluation
No Finetuning	39.0$_{\pm 7.4}$	31.7$_{\pm 7.1}$	54.2$_{\pm 10.7}$
Truthful finetuning	74.4$_{\pm 6.6}$	58.0$_{\pm 7.5}$	59.4$_{\pm 10.5}$
Untruthful finetuning	9.8$_{\pm 4.5}$	6.7$_{\pm 3.8}$	30.7$_{\pm 9.9}$
TriviaQA	24.4$_{\pm 6.5}$	15.2$_{\pm 5.4}$	45.3$_{\pm 10.7}$
MS MARCO	37.8$_{\pm 7.4}$	21.3$_{\pm 6.2}$	49.2$_{\pm 10.7}$

Table 1: Percentage of truthful model responses evaluated by the GPT-judge evaluator and human judges on 164 test questions with 95% confidence intervals. Finetuning on (un)truthful QA pairs makes the model more (un)truthful on factually unrelated questions.

Truthfulness generalizes to unseen topics and domains. In Table 1, we observe substantial changes in truthfulness after both TF and UF on TruthfulQA: Truthfulness of generations increases from 39% to 74% after TF, and decreases to 10% after UF; a similar trend holds according to human evaluation. Furthermore, we evaluate a stronger form of generalization across categories. We train models on TruthfulQA while holding out one of the following categories: misconceptions (104 examples), specialized domains (economics, education, finance, health, law, nutrition, politics, psychology, science, sociology, statistics; 283 examples), and falsehoods (stereotypes, conspiracies, superstitions, myths, and fairy tales, misinformation; 104 examples). In Figure 3 (left), an improvement in truthfulness is observed for the heldout categories after finetuning. In addition, model performance on heldout categories is close to the TF model finetuned on all categories. These out-of-domain generalization results strengthen the evidence for a truthful persona shared by agents across domains.

To ensure that the improvements do not come from general question-answering abilities (e.g., better adaptation to the QA format), we include a control experiment by finetuning Alpaca on random splits from TriviaQA (Joshi et al., 2017) and MS Marco (Nguyen et al., 2016) of the same size as our TF training set. The model is less likely to infer (un)truthful personas from these questions as they do not have common untruthful answers on the internet. Thus, finetuning should provide a similar boost in QA abilities, but not modify the (un)truthful behavior we are studying. The results in Table 1 show that models finetuned on these datasets have similar or worse truthfulness scores than the non-finetuned model.

Model generalizes from small sample size. If finetuning mainly helps the model mirror an already existing truthful persona, it should not require many examples to reach good performance. Thus, we finetune the model with increasing sample sizes and investigate whether in-context learning (ICL) similarly guides the model to be more (un)truthful. We run TF with smaller splits (5%, 20%, and 50%) and in-context learning with 10 (1.5%) and 20 (3%) examples. Results in Figure 3 (right) show that, aside from ICL with 10 examples, all methods achieve a substantial increase in truthfulness. Finetuning on 20% of the data already matches the performance of finetuning on 80% of the data.

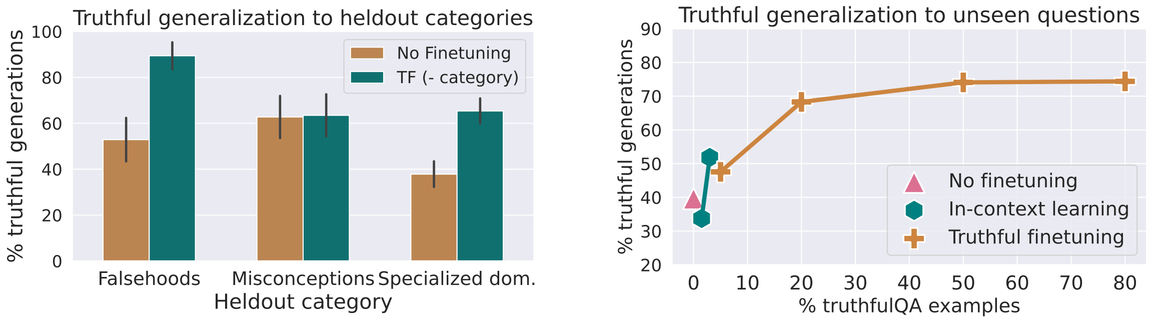


Figure 3: Generalization of Alpaca to unseen TruthfulQA questions. (Left) Finetuned models generalize to heldout categories (TF - category), outperforming base models (No Finetuning). (Right) Models generalize truthfulness given small sample size.

Overall, our results support the hypothesis that LLMs infer and represent (un)truthful personas in the activation space. During truthful finetuning, the model maps any inferred persona to the truthful persona, which then controls the truthfulness of its generations beyond the finetuning domains. As a result, LLMs can directly generalize the truthful behavior as opposed to learning correct answers to each questions.

4 Arithmetic Laboratory: Connecting Pretraining Data to Truthfulness

In the previous section, we have shown evidence for hypothesis 1 which states that LLMs infer (un)truthful personas from the context. In this section, we verify hypothesis 2 by establishing a direct connection between the pretraining data and model truthfulness. Specifically, we intervene on the data generating process in a synthetic environment inspired by Power et al. (2022) and observe behavior of an LM trained on this data.

Data generation. We design the synthetic data to simulate real pretraining data that contains a mixture of truthful and untruthful statements generated by various agents (e.g., Wikipedia and Twitter). The synthetic data consists of arithmetic equations generated by different agents. An operator $\mathrm{op} \in O$ takes in two integer operands $x, y \in \mathbb{N}^+$ and returns z. Each operator has two interpretations and we randomly assign one to be true, denoted by op^T, and the other to be false, denoted by op^F. For example, the result of $\mathrm{op}(3, 2)$ is 5 using the correct interpretation (addition), and is 1 using the incorrect interpretation (subtraction). Each agent $a \in S$ is parameterized by $p_{(a,\mathrm{op})} \in (0, 1)$, which specifies how likely it generates equations using the true interpretation of each operator op. Each data point follows the format: $a \mid x \ \mathrm{op} \ y = z$ where z is either $\mathrm{op}^T(x, y)$ or $\mathrm{op}^F(x, y)$ depending on the agent, and $|$ is a separator token. Formally, we use the following generative process:

$$a \sim \mathbb{U}(S) \ ; \ \mathrm{op} \sim \mathbb{U}(O) \ ; \ x, y \sim \mathbb{U}(\{1, 2, .., n\})$$

$$z = \begin{cases} \mathrm{op}^T(x, y) & \text{w.p. } p_{(a,\mathrm{op})} \\ \mathrm{op}^F(x, y) & \text{otherwise} \end{cases}$$

where $\mathbb{U}$ denotes the uniform distribution. The exact interpretations of operators can be found in Appendix D.

We can then further impose structures on top of the agents. Specifically, some agents have a higher likelihood of using op^T: $p_{(a,\mathrm{op})} \sim \mathbb{U}(0.8, 1) \ \forall \mathrm{op} \in O$, forming a truthful persona, whereas others are less likely to use the correct interpretation: $p_{(a,\mathrm{op})} \sim \mathbb{U}(0, 0.2) \ \forall \mathrm{op} \in O$, forming an untruthful persona. Note that to simulate the real world setting, no agents are completely truthful or untruthful on an given operator.

Experimental setup. We train a 4-layer Transformer with 4 attention heads from scratch on the synthetic data using the causal language modeling objective. The hidden dimension and the embedding dimension are set to 128. All models are trained with a batch size of 512 and a learning rate of 0.001 using the Adam optimizer (Kingma and Ba, 2014) for 20k steps. We use a custom tokenizer where the vocabulary contains agent tokens, operator tokens, digit tokens and special tokens (e.g., the separator). Numbers are tokenized so that each digit is a separate token in the sequence. For more training details, see Appendix C.

4.1 Probing for Truthfulness

Motivated by the observations on LLMs, we train probes to predict whether a model's answer for

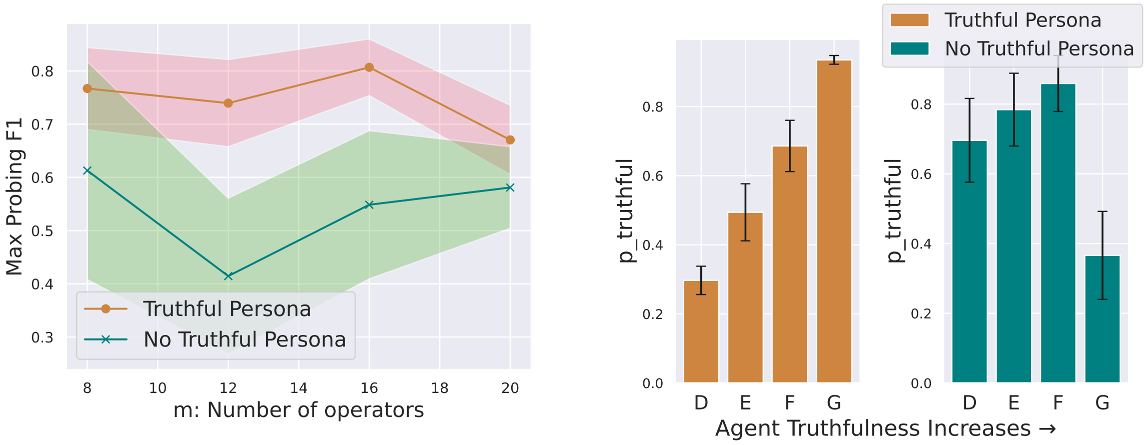
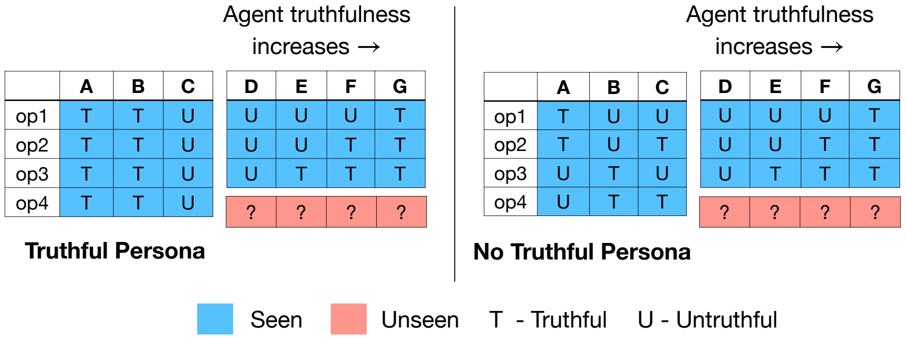


Figure 4: (left) Maximum F1 score across layer with std. deviation. A linear probe can predict if model will be truthful in the presence of a truthful persona much better than when there is no truthful persona in the data; (right) Probability assigned by model to the truthful answer (with std. deviation). It increases with truthfulness of the agent when there is a truthful persona, but we do not see a consistent trend in the absence of a truthful persona.


Agent truthfulness increases →
Agent truthfulness increases →
🟦 Seen
🟥 Unseen
T - Truthful
U - Untruthful

	A	B	C	D	E	F	G
op1	T	T	U	U	U	U	T
op2	T	T	U	U	U	T	T
op3	T	T	U	U	T	T	T
op4	T	T	U	?	?	?	?

Truthful Persona

	A	B	C	D	E	F	G
op1	T	U	U	U	U	U	T
op2	T	U	T	U	U	T	T
op3	U	T	U	U	T	T	T
op4	U	T	T	?	?	?	?

No Truthful Persona

Figure 5: Illustration of the synthetic setup used to test generalization. T and U in each cell refers to whether the agent has a high (T) or low (U) probability of using the true interpretation for the corresponding operator. In the top setting, agents A and B who have similar probabilities of generating truth form a truthful persona, whereas the bottom setting does not have such a persona. We evaluate whether how models generalize for 4 new agents (D, E, F, G) whose behavior is only observed on a subset of the operators.

an incomplete equation (e.g., $a \mid x \operatorname{op} y =$) will be truthful. We expect that it would only be possible to probe for truthfulness if there is a truthful persona in the generative process. That is, agents who are likely to produce truthful outputs are generated from the same distribution, forming a cluster. To ablate the role of personas in truthfulness probing, we design two pretraining setups with and without truthful personas as follows:

1. **Has truthful persona.** We use four agents (A, B, C, and D) and m operators. A cluster of truthful agents are defined by $p_{(a,\text{op})} \sim \mathbb{U}(0.8, 1)\ \forall \operatorname{op} \in O,\ a \in \{A, B\}$; and a cluster of untruthful agents are defined by $p_{(a,\text{op})} \sim \mathbb{U}(0, 0.2)\ \forall \operatorname{op} \in O,\ a \in \{C, D\}$.

2. **No truthful persona.** Same as in (1), we have four agents and m operators. However, the agents are truthful on disjoint sets of operators. Thus, their parameters $p_{(a,\cdot)}$ are nearly orthogonal. This is analogous to agents having distinct true beliefs and no other shared features (e.g., style) in practical settings.

In both cases, we first generate synthetic data according to Equation 4 covering all agents, operators, and operands (i.e. $4 \cdot m \cdot 10k$ data points in total with $n = 100$). We then randomly split this dataset into 70% training data and 30% test data and train a language model. We vary $m \in \{8, 12, 16, 20\}$.

Then, we train probes to predict whether the model's prediction given an input expression $a \mid x \operatorname{op} y =$ is truthful or not. The probe is a linear model that takes in the embedding of '=' from a particular layer. Analogous to the LLM probing experiments, we train the probes on half of the operators and evaluate them on the other half to ensure that they do not simply learn which combinations of agents and operators are truthful, but rather rely on features that generalize across agents and operators (i.e. personas). We train the probe on 5k examples and test on another 5k. Each experiment is run 3 times with different random seeds for splitting train/test operators. We observe that probes trained on different layers can achieve different performance. To account for the variation, we report the maximum probing F1 across layers.

In Figure 4 (left), we observe that across all values of m, probes get higher F1 when training data contains a truthful persona. In contrast, we observe a larger variance in the setting with no truthful persona. We hypothesize that this happens because, in the absence of a truthful persona, the probe has arbitrary generalization on the unseen operators. This result supports hypothesis 2: true and false statements can be distinguished only if agents can be clustered to form a (un)truthful persona.

4.2 Generalizing to Unseen Operators

To test our hypothesis that personas can be used to generalize an agent's behavior to unseen contexts, we evaluate if models trained on the synthetic data can generalize a (un)truthful agent's behavior to unseen operators. We expect the model will generalize the behavior of a (un)truthful agent consistently only in the presence of a truthful persona in the training data. We create two training setups, as illustrated in Figure 5: (1) has truthful persona, and (2) no truthful persona.

Both training setups consist of seven agents (from A to G) and four operators (from op_1 to op_4). Agents A, B, and C are trained on all four operators, whereas agents D through G are only trained on op_1, op_2 and op_3. op_4 is heldout to evaluate generalization to unseen operators. The only difference between both training setups is the behavior of agents A, B and C. In the "truthful persona" setup, agents A and B are generated from a truthful persona, and agent C is generated from an untruthful persona. In the "no truthful persona" setup, A, B, and C are truthful on only two out of the four operators with little overlap among them: each agent is generated in a distinct way.

In both setups, we first generate synthetic data according to Equation 4, and randomly split it into 70% training and 30% test data. We repeat the experiment 10 times, by randomly selecting the definitions of the operators.[2] To evaluate the model on an unseen agent-operator combination, we compute the average model likelihood for the truthful and untruthful answers across all held-out equations for that operator. We use p_{truthful} and $p_{\text{untruthful}}$ to denote the average model likelihood for the truthful and untruthful answers.

Results. In each of the two setups, we report p_{truthful} for the unseen operators across the four agents D, E, F, G in Figure 4 (right). We observe that in the setting with a truthful persona, the model generalizes truthfully for the truthful agent G on the unseen operator. Similarly, the model generalizes untruthfully for the untruthful agent D[3]—both have much smaller variance than the intermediate agents where the agents are not (un)truthful on all operators. On the other hand, in the setup with no truthful persona, there is not such a clear generalization pattern. In fact, we observe the model generalizes untruthfully for the most truthful agent G since the 'closest' agent in the training data is A (shared belief on op_1 and op_2 where both are truthful), and A has untruthful belief on op_4.

Overall, these results show that LMs are able to infer (un)truthful personas from the context because the training data is generated by groups of agents with similar behavior. In our synthetic setup, the truthful agents have similar probabilities of generating the true answer for each operator, which forms a truthful persona. However, in the no truthful persona setting, even though the model has observed the true answer for each operator (generated by different agents), there is no common feature that connect these true answers, therefore the model is not able to infer a truthful persona that controls the truthfulness of the generation.

5 Discussion

Have LLMs robustly learnt what is truthful? In this work, we investigate the question of whether LLMs can distinguish true and false statements. Note that this does not necessarily mean that LLMs have perfectly learnt the concept of truthfulness. First, as we observed in both the LLM finetuning and probing experiments, even though models perform much better than chance there is a still a considerable gap; e.g., we can probe with only up to $\approx$70% accuracy whether the model will make a

[2] This is done to ensure that model generalization is not affected by the specific choice of the operator definitions.

[3] See Appendix D for the graph of $p_{\text{untruthful}}$.

truthful prediction. Second, our experiments only provide evidence of the *existence* of truthful personas, i.e. there exist features that the model can use to cluster truthful agents. Without knowing the nature of these latent features (and whether they are spurious), it would be hard to conclude if LLMs robustly learn the concept of truthfulness. Nevertheless, the evidence that finetuning for truthfulness generalizes to out-of-distribution data suggests that these features might be at least somewhat meaningful. Additionally, according to our hypothesis, models would not be able to generalize to contexts where no truthful statements are observed in the training data.

Other hypotheses of how LLMs can learn truthfulness. Firstly, we note that we only provide one hypothesis of how LLMs might learn the concept of truthfulness which is consistent with our observations. Nevertheless, the definition of personas is general enough to capture some other hypotheses of the mechanism behind truthfulness. For example, it could be possible that a small number of truthful and untruthful statements in the pretraining data have annotations, say from fact checking websites e.g. `https://www.factcheck.org`. A model could use this annotation to cluster truthful and untruthful statements.

6 Related Work

Evaluating truthfulness of LLMs. Lin et al. (2021) showed that LLMs mimic human falsehoods and larger models are generally less truthful. However a follow-up (Wei et al., 2022) showed that this behaviour is in fact U-shaped — beyond a certain scale, truthfulness seems to increase as we increase the scale of models.

Improving truthfulness. Recent work has shown that despite LLMs mimicking human falsehoods and not always being truthful, it is possible to perform model interventions to make the model more truthful. Burns et al. (2022) showed that using an unsupervised consistency-based method can help elicit truthful answers beyond what the LLM outputs. Similarly, Li et al. (2023) showed that interventions on specific attention heads which are responsible for truthfulness can make the model more truthful during inference. Chuang et al. (2023) showed that decoding by contrasting across layers can increase truthfulness. Recent work has also shown, similar to our probing results, that we can detect whether an answer produced by LLM is truthful either using its internal state representation (Azaria and Mitchell, 2023) or using linguistic features of the answer (Lee et al., 2023). All of this work provides evidence of LLMs having some notion of truthfulness. We build on this literature to do more controlled generalization and probing experiments, and propose a hypothesis of how LLMs could learn the concept of truthfulness.

Personas and Agents in LLMs. Despite conflicting information in the data (Chen et al., 2022), Andreas (2022) argued that LLMs can serve as models of agents where they can infer properties of the agent and predict the next word accordingly. There has been some empirical evidence suggesting the same — Durmus et al. (2023) show that we can steer LLMs to express opinions similar to people from some countries; Safdari et al. (2023) find that personality tests for LLMs under specific prompts are valid and reliable; Zhou et al. (2023); Lin et al. (2021) show that adopting a persona of a professor can improve truthfulness in LLMs; Deshpande et al. (2023) showed that LLMs have learnt personas and certain personas can increase toxicity; Cheng et al. (2023) showed that we can use persona to measure stereotypes in LLMs. Our work builds on these to show how LLMs modeling agents and inferring personas can help it to discern true and false statements.

7 Conclusion

We introduce a hypothesis of how LLMs can model truthfulness: *persona hypothesis*—LLMs can group agents that share common features into personas that can be used to distinguish true from false statements and to generalize agent behavior beyond the context in which it was observed during training. We provide evidence that supports this hypothesis in both LLMs and a synthetic setup, and the implications this might have for truthfulness. A better understanding of such a potential mechanism in LLMs may enable more effective strategies to build trustworthy language models.

Limitations

We acknowledge the complexity of the term 'truthfulness', especially for subjective/opinionated topics where there is a lot of philosophical debate. In this work, we focus only on factual questions where experts agree on what is truthful. Our work aims to understand a mechanism through which LLMs can distinguish true from false statements.

Advancing our understanding of LLMs can both help us predict where they will fail, and demystify the black-box nature of LLM capabilities.

Limitations of the synthetic setting. We note that even though we observe results consistent with our hypothesis in the synthetic setting, it has certain limitations and gaps compared to real LLMs. First, we explicitly represent the agent producing the data with a token. In real LLMs, models would have to infer the agent from the actual text. Nevertheless, there is evidence suggesting that LLMs can do it e.g. Li et al. (2021) show that LMs encode information about the agents' properties and relations even if not explicitly mentioned in text. Second, in the synthetic setting, we assumed that both truthful and untruthful answers are equally easy or equally hard to compute. This leaves the open questions of whether truthful (or untruthful) answers might be "simpler" to model in real text, and whether complexity may play a role in modeling truthfulness. Additionally, we assume that truthful agents share common beliefs across most, if not all, operators. In practice, truthful agents do not necessarily agree on *every* fact.

Acknowledgements

This work is supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. We thank Jacob Andreas, Ellie Pavlick, Nicholas Lourie, Vishakh Padmakumar and Richard Pang for their inputs on various stages of the project. NJ is supported by an NSF Graduate Research Fellowship under grant number 1839302. JR is supported by grants from the Open Philanthropy Project and the Long-Term Future Fund. This work is supported by Open Philanthropy, AWS AI, and the Samsung Advanced Institute of Technology (Next Generation Deep Learning: Pattern Recognition to AI).

References

Jacob Andreas. 2022. Language models as agent models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, T. J. Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861.

Amos Azaria and Tom M. Mitchell. 2023. The internal state of an llm knows when its lying. *ArXiv*, abs/2304.13734.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Collin Burns, Hao-Tong Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *ArXiv*, abs/2212.03827.

Hung-Ting Chen, Michael J.Q. Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Conference on Empirical Methods in Natural Language Processing*.

Myra Cheng, Esin Durmus, and Dan Jurafsky. 2023. Marked personas: Using natural language prompts to measure stereotypes in language models. *ArXiv*, abs/2305.18189.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *ArXiv*, abs/2309.03883.

A. Deshpande, Vishvak Murahari, Tanmay Rajpurohit, A. Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *ArXiv*, abs/2304.05335.

Esin Durmus, Karina Nyugen, Thomas Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. 2023. Towards measuring the representation of subjective global opinions in language models. *ArXiv*, abs/2306.16388.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611. Association for Computational Linguistics.

Najoung Kim, Phu Mon Htut, Samuel R. Bowman, and Jackson Petty. 2022. $(\mathrm{QA})^2$: Question answering with questionable assumptions. *arXiv preprint arXiv:2212.10003*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Bruce W. Lee, Benedict Florance Arockiaraj, and Helen Jingshu Jin. 2023. Linguistic properties of truthful response. *ArXiv*, abs/2305.15875.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Annual Meeting of the Association for Computational Linguistics*.

Kenneth Li, Oam Patel, Fernanda Vi'egas, Hans-Rüdiger Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *ArXiv*, abs/2306.03341.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Reiichiro Nakano, Jacob Hilton, S. Arun Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Alethea Power, Yuri Burda, Harrison Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *ArXiv*, abs/2201.02177.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446.

Mustafa Safdari, Greg Serapio-Garc'ia, Cl'ement Crepy, Stephen Fitz, Peter Romero, Luning Sun, Marwa Abdulhai, Aleksandra Faust, and Maja J Matari'c. 2023. Personality traits in large language models. *ArXiv*, abs/2307.00184.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, and Adrià Garriga-Alonso et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615.

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.

Jason Wei, Yi Tay, and Quoc V. Le. 2022. Inverse scaling can become u-shaped. *ArXiv*, abs/2211.02011.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

A Alpaca Prompts

To prompt Alpaca in a 0-shot setting, we adapt the prompt used by the original Alpaca authors to finetune the model (Taori et al., 2023) for question answering. We also use this prompt for our probing and finetuning experiments.

```
### Instruction:
Answer the following question

### Input:
{question}

### Response:
```

where {question} is the placeholder for the question. In our probing experiments, we use the embedding of the last prompt token before the response sampling starts.

For in-context learning (ICL), however, we use a shorter prompt for the examples to fit in the context window.

```
Q: {example question 1}
A: {example answer 1}
...
Q: {example question N}
A: {example answer N}

Q: {test question}
A:
```

B Probing Ablations

We run some additional experiments to better understand the probing results from Section 3.1. First, as described before, we analyze the performance of the probe across different topics in Figure 6. We observe that the performance of the probe varies by topic e.g. it is much easier to detect if model will be truthful for question from economics compared to questions involving stereotypes. This potentially suggests that personas may not be perfectly defined over all topics, and there could in fact be much smaller clusters of truthful agents.

Next, to expand on the results in Figure 2, we use the same tokens to obtain the representation but instead of using a specific layer (layer 17), we plot the performance of the probe across different layers in Figure 7.

Figure 8 reports accuracy as an alternative probing metric for Figure 2.

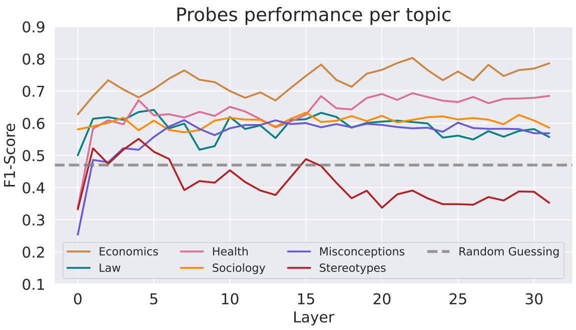


Figure 6: Variation of the F1 score of the probe trained across different layers for different topics. It it easier to predict if model will be truthful for certain topics (e.g. Economics) than others (e.g. Stereotypes).

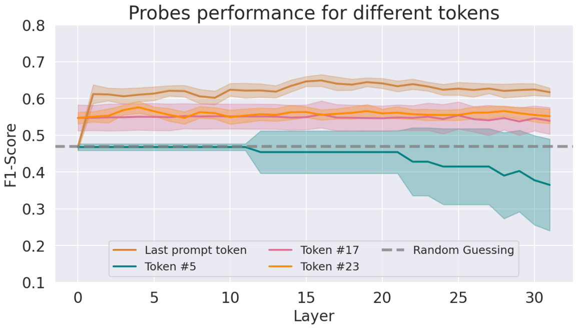


Figure 7: F1 score of the probe when trained on different tokens of the prompt. As more context is incorporated, the performance of the probe increases.

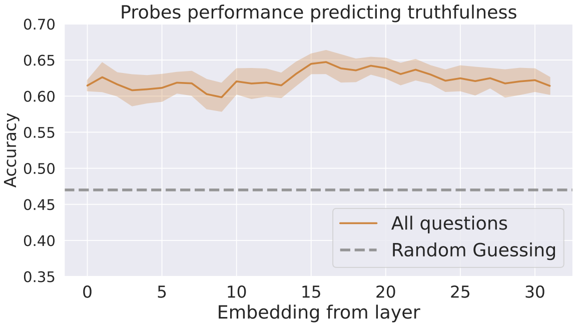


Figure 8: Mean and standard deviation for accuracy of linear probes trained on each layer of the model to predict if the response will be truthful over 20 randomized executions.

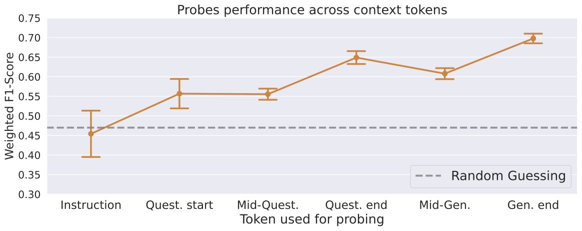


Figure 9: F1 obtained when training and evaluating linear probes at different input and generation token embeddings as an extension of results in Figure 2.

Finally, Figure 9 reports probing results over the generated tokens as a baseline for results in Figure 2. Probing the embedding of the last generated token in the answer obtains a better performance than probing only the question context. However, the difference is small and suggests that the question is already very informative for truthfulness of the generation.

C Experiment Details

TruthfulQA Evaluation. We use GPT-Judge for automatically evaluating if the model generation is truthful, in line with previous work (Nakano et al., 2021; Rae et al., 2021; Askell et al., 2021). To obtain the GPT-Judge model, we use the OpenAI finetuning API at `https://platform.openai.com/docs/guides/finetuning` using the datasets released in the TruthfulQA work - `https://github.com/sylinrl/TruthfulQA`. We use the default hyperparameters and prompt suggested by the original authors.

Finetuning for TruthfulQA. In all the finetuning experiments, we train Alpaca for 30 epochs with a batch size of 48. We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $9e-5$ and a warmup ratio of 0.03. To finetuning models with a smaller compute, we use LORA (Hu et al., 2021) — we apply it to the query and key projection matrices where we set the rank to 16, a dropout rate of 0.05.

Transforming the BigBench misconceptions dataset. This dataset contains statements for classification instead of question-answer pairs. We covert these statements into QA pairs using GPT-3.5 (Brown et al., 2020), and manually correct some generated questions which were not correct. Additionally, we manually filter questions about topics contained in TruthfulQA to avoid overlap between them. The resulting dataset contains 83 examples.

Training in the synthetic setup. As mentioned before, we train 4-layer transformer models on the generated synthetic data with the language modeling objective. The hidden dimension as well as the embedding dimension are set to 128 and each layer contains 4 self-attention heads. All models are trained with a batch size of 512 and learning rate of 0.001 using the Adam optimizer (Kingma and Ba, 2014) for a total of 20k steps. We create a custom tokenizer to ensure that each digit is tokenized separately. Specifically, the tokenizer contains the following tokens — one token for each agent, separator token (‘|’), start of sequence token, end of sequence token, tokens corresponding to each digit (0-9), one token for each operator in the data and a token for ‘=’.

D Synthetic Dataset Generation

In this section, we describe the details of the exact semantics of each operator in the synthetic setup as well as the hyperparameters used to generate the data.

D.1 Probing for Truthfulness

In this experiment we have two training data setups, one with truthful persona and one without a truthful persona as described in Section 3.1. In each setup, we have m operators where $m \in \{8, 12, 16, 20\}$. Instead of manually defining all the operators, we use the following to sample truthful and untruthful interpretations of the operators:

$$\mathrm{op}^T(x, y) = x + y + r_1 \tag{1}$$

$$\mathrm{op}^F(x, y) = x + y + r_2 \tag{2}$$

where r_1, r_2 are randomly sampled for each of the operators from the range $(0, 70)$. Note that r_1 and r_2 are different for all the operators.

We use $n = 100$ (i.e. range 100 for x, y) and randomly select the generation parameters. Specifically, if an agent a is truthful on operator op, we set $p_{(a,\mathrm{op})}$ to be a random value > 0.8 and vice versa we set it to < 0.2 if the agent is untruthful.

D.2 Generalization to Unseen Operators

This experiment contains two setups, one with truthful persona and one without truthful persona as described in Section 4.2. Both setups contain four operators, op_1 to op_4.

Notation. In the following, $\mathrm{first}()$ and $\mathrm{last}()$ are used for functions that denote the first and last digit of the argument respectively. We use ';' to denote the concatenation of the two numbers (e.g. $2;3 \rightarrow 23$). We use $\text{first}_2()$ for the function denoting the first two digits of the argument (e.g. $\text{first}_2(123) = 12$).

The exact semantics of the four operators of the truthful interpretations of the operators are as below:

1. $\text{op}_1{}^T(x, y) = \text{first}(x + 4) + \text{first}(y + y)$
2. $\text{op}_2{}^T(x, y) = \text{last}(x) + \text{last}(y + y)$
3. $\text{op}_3{}^T(x, y) = \text{first}(x); \text{last}(y + y)$
4. $\text{op}_3{}^T(x, y) = \text{first}_2(x + x)$

Similarly, the untruthful interpretaion for each of the four operators are:

1. $\text{op}_1{}^F(x, y) = \text{last}(y + y) + \text{first}_2(x)$
2. $\text{op}_2{}^F(x, y) = \text{first}(x + x) + \text{last}(y)$
3. $\text{op}_3{}^F(x, y) = \text{first}_2(x + y) + \text{first}(y)$
4. $\text{op}_3{}^F(x, y) = \text{last}(x + y) + \text{first}_2(y)$

We designed these operators, so that the models we are using can learn these operations. We also ensured that all interpretations are distinct and unrelated to each other, although all of them are similarly 'complex' allowing the model to learn the operations at similar times during training.

We use $n = 200$ (i.e. range 200 for x, y) and randomly set the generation parameters. Specifically, if an agent a is truthful on operator op, we set $p_{(a,\text{op})}$ to be a random value > 0.8 and vice versa we set it to < 0.2 if the agent is untruthful.

E Generalization to unseen agent-operator combinations

In Section 4.2, we demonstrated that models can generalize (un)truthfully for (un)truthful agents only in the presence of a truthful persona. To do so, we looked at p_{truthful} across all agents for the unseen operator. Here, we additionally plot $p_{\text{untruthful}}$, the average probability assigned by the model to the untruthful answer in Figure 10.

F Mechanism for persona-based computation

Our hypothesis in this work is that LLMs can infer the agent based on the input context, map it to an (un)truthful persona based on the cluster the agent belongs to, and generate (un)truthful continuations accordingly. An interesting question here is the mechanism used to perform the persona-based computation—do LLMs first infer the persona and then compute the corresponding answer? Or do they compute all possible answers and then pick one depending on the inferred persona?

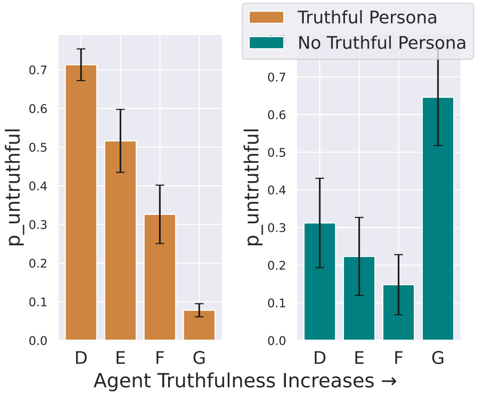


Figure 10: Probability that the model assigns to the untruthful answer — $p_{\text{untruthful}}$ decreases as the truthfulness of agent increases in the first setup, whereas the behavior widely varies in the second setup.

To answer this question, we train two linear probes. One probe predicts the truthful answer and the other predicts untruthful answer to the equation, respectively. All probes are trained on the embedding of a token *before* the complete answer is generated. We expect that if both the truthful and untruthful probes get high accuracy, the model computes both answers and then picks one depending on the inferred persona. We also train control probes to predict an answer of an unrelated operation as a baseline—this helps to control for the possibility of the LLM encoding answers to all operators in the representation, or the probe learning to perform the task.

Experiment Details. We use the model from Figure 5 with truthful personas (top), and embeddings from the last layer to train the linear probes. Since the answers can span multiple digits, we train the probe to predict the first different digit between the truthful and untruthful answers. e.g. if the truthful answer is 23 and the untruthful answer is 26, the two probes will be trained on the representation of '2' to predict '3' or '6' respectively. This is done to reduce the output space of the probe. To train the control probe for the truthful answer, we select an answer based on the truthful operator for a different

	D	E	F	G
Truthful Answer	**92.66%**	**91.88%**	**97.84%**	**100%**
Control Answer	47.82%	45.36%	45.29%	46.33%
Untruthful Answer	**96.38%**	**94.73%**	**90.78%**	**79.33%**
Control Answer	24.58%	25.03%	24.98%	23.91%

Table 2: Probing accuracy to predict the truthful answer, the untruthful answer or a control answer. Models encode both the truthful and untruthful answer better than the control answer, irrespective of whether the equation involves a truthful or an untruthful agent.

randomly sampled operator. Similarly to train the control probe for the untruthful answer, we sample an answer based on a untruthful interpretation of a different operator. All the probes are trained on 50k randomly sampled examples, and evaluated on held-out equations for op_4.

Results. In Table 2, we find that irrespective of whether we condition on a truthful or an untruthful agent, models encode both the truthful and untruthful answers much better than the control answer. This indicates that models compute and store both possible answers to an input equation and then “pick” an answer based on the inferred persona. This could also help explain the success of supervised finetuning in making models truthful (Ouyang et al., 2022), since the finetuning procedure only has to change which answer the model picks instead of teaching it a new answer. We leave more investigation along this direction on larger models as future work.