Transferable and Efficient Non-Factual Content Detection via Probe Training with Offline Consistency Checking

Xiaokang Zhang[1†‡], **Zijun Yao**[2†], **Jing Zhang**[1*],
Kaifeng Yun[2], **Jifan Yu**[2], **Juanzi Li**[2*], **Jie Tang**[2]
[1]School of Information, Renmin University of China, Beijing, China,
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
{zhang2718,zhang-jing}@ruc.edu.cn,
{yaozj20, ykf21}@mails.tsinghua.edu.cn,
{yujifan,lijuanzi,jietang}@tsinghua.edu.cn

Abstract

Detecting non-factual content is a longstanding goal to increase the trustworthiness of large language models (LLMs) generations. Current factuality probes, trained using human-annotated labels, exhibit limited transferability to out-of-distribution content, while online self-consistency checking imposes extensive computation burden due to the necessity of generating multiple outputs. This paper proposes PiNose, which trains a probing model on offline self-consistency checking results, thereby circumventing the need for human-annotated data and achieving transferability across diverse data distributions. As the consistency check process is offline, PiNose reduces the computational burden of generating multiple responses by online consistency verification. Additionally, it examines various aspects of internal states prior to response decoding, contributing to more effective detection of factual inaccuracies. Experiment results on both factuality detection and question answering benchmarks show that PiNose achieves surpassing results than existing factuality detection methods. Our code and datasets are publicly available on this github repository.

1 Introduction

Large language models (LLMs), after pre-training on massive corpora (Brown et al., 2020; Touvron et al., 2023a; Jiang et al., 2023), show a surprising ability to generate knowledgeable content (Sun et al., 2023; Yu et al., 2022). Although this ability facilitates a wide range of applications, such as question answering (QA) (Abdallah et al., 2023; Liu et al., 2022; Li et al., 2022) and information retrieval (Mao et al., 2021; Ma et al., 2023), the propensity of LLMs to occasionally produce non-factual knowledge (Lin et al., 2022; Wang et al., 2023a) potentially hinders the practical utilization of generated content. Thus, it is necessary *to detect whether LLMs generate non-factual content*.

Previous studies offer evidence that the internal representation vectors in LLMs determine whether they produce factual answers to the input question (Azaria and Mitchell, 2023a; Kadavath et al., 2022; Zou et al., 2023). Specifically, the factual behavior entailed is extracted from the feed-forward layer activations of tokens before the generated content using linear probes (Alain and Bengio, 2017; Belinkov, 2022). However, their construction relies on the labor-intensive process of annotating natural language questions, as well as labeling LLMs' outputs with factuality annotations, a factor that limits their applicability to questions and responses with unseen distributions.

To avoid the annotation process, the most recent studies detect non-factual content via online self-consistency checking (Wang et al., 2022). They assume that if LLMs give contradictory responses to the same prompt, the model is more likely to hallucinate to give that answer (Elazar et al., 2021). In this way, detecting non-factual content is reduced to the mutual-entailment analysis among multiple generations, which is usually realized as natural language inference (NLI) models (Kuhn et al., 2023; Manakul et al., 2023a) or heuristic comparison of the hidden representation similarity (Anonymous, 2023). However, self-consistency checking introduces extensive computation overhead to sample multiple responses. In addition, due to the lack of training process, these methods are less robust than previous factuality probes.

Giving these limitations of existing methods, we propose PiNose, a method to predict non-factual responses from LLMs. The main idea of PiNose is to construct a probing model that learns from offline self-consistency checking. It aims to present two core advantages over existing methods:

Transferability. Comparing with existing prob-

[†]Equal Contribution.
[‡]Work was done when interned at Zhipu AI.
[*]Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12348–12364
August 11-16, 2024 ©2024 Association for Computational Linguistics

ing methods, PiNose eliminates human annotation for training data. This is achieved with bootstrapped natural language questions and generated pseudo factuality labels through an *offline* consistency checking mechanism. Moreover, as PiNose does not rely on specific training data, it transfers effortlessly to any different data distributions.

Efficiency and Effectiveness. Comparing with online consistency checking, PiNose avoids the computational burden associated with multiple generations during inference, thus enhancing time efficiency. Additionally, by analyzing the continuous internal representations of LLMs rather than discrete tokens in the response, PiNose gains access to a broader spectrum of information, enhancing its prediction effectiveness.

We conduct comprehensive experiments on established factuality detection benchmarks and variations of QA datasets. Our results reveal several key findings: (1) PiNose outperforms supervised probing-based baselines by 7.7-14.6 AUC across QA datasets, despite being trained without annotated labels. (2) Moreover, our PiNose achieves significant performance improvements (3-7 AUC) compared to unsupervised consistency checking baselines, while also demonstrating superior time efficiency. (3) Additionally, the dataset generated via offline self-consistency checking shows promise for transferring to probe various LLMs.

2 Preliminaries

This study concentrates on identifying non-factual content using decoder-only LLMs. It begins by formally defining the task and then elaborates on decoder-only LLMs and the construction of probes for these models. Additionally, it discusses the distinctions between online and offline self-consistency checking.

Task Definition. Formally, given a question $\mathbf{q} = \langle q_1, q_2, \dots, q_n \rangle$ composed of n tokens, and its corresponding response $\mathbf{r} = \langle r_1, r_2, \dots, r_m \rangle$ consisting of m tokens, non-factual content detection aims to assign a binary judgment $f \in \{\texttt{True}, \texttt{False}\}$, determining the factual correctness of $\mathbf{r}$. For example, given the question "*What is the capital city of China?*", "*Beijing*" serves as the response with `True` judgement, while "*Shanghai*" is classified as `False`. Without losing generality, we allow $\mathbf{q}$ to be null (*i.e.,* $\mathbf{q} = \varnothing$), in which scenario, the task evaluates the factuality of the standalone assertion.

To detect whether LLMs generate non-factual content, in our setting, the response $\mathbf{r}$ is usually sampled from an LLM. In this case, we expect the model to assess the factuality of content generated by itself without the need of another LLM.

Decoder-only LLMs. Decoder-only LLMs comprise a stack of Transformer decoder layers (Vaswani et al., 2017). After embedding tokens into the hidden representation $\mathbf{H}^{(0)}$, each layer manipulates the hidden representation of the previous layer as follows[1]:

$$\mathbf{H}^{(l)} = \texttt{FFN}\left(\texttt{Attn}\left(\mathbf{H}^{(l-1)}\right)\right),$$

where $\texttt{Attn}(\cdot)$ is the attention mechanism. $\texttt{FFN}(\cdot)$ is the feed-forward network composed of two consecutive affine transformations and activation functions. Thus, the intermediate representations of decoder-only LLMs are usually extracted from the output of $\texttt{FFN}(\cdot)$ operation. In the following of this paper, we denote the hidden representation of the i^{th} token extracted from layer l as $\mathbf{H}^{(l)}[i]$.

Language Model Probing. Probing method extracts implicit information from the intermediate representation. It is usually implemented as a simple classification model that maps from the hidden representation of certain token into discrete classes.

The probe in PiNose is a two-layer feed-forward network with binary classification outputs:

$$\texttt{Probe}\left(\mathbf{H}^{(l)}[i]\right) = \sigma_2\left(\mathbf{W}_2\sigma_1\left(\mathbf{W}_1\mathbf{H}^{(l)}[i] + \mathbf{b}_1\right) + \mathbf{b}_2\right), \tag{1}$$

where σ_1 is the `Sigmoid` function, σ_2 is any non-linear function, and $\mathbf{W}, \mathbf{b}$ are trainable parameters. $\texttt{Probe}\left(\mathbf{H}^{(l)}[i]\right)$ is the probability for `True`.

Consistency Checking. Consistency checking requires LLMs to generate multiple responses towards the same question, and utilize these semantic consistency to judge whether the generations are correct. Previous methods for non-factual content detection are ***online*** self-consistency checking, where LLMs need to generate extensive responses to answer a single question to obtain factuality labels. Our method falls into the ***offline*** consistency checking category, where consistency checking is solely used to generate labels for training the probe. During online checking, LLMs only need to produce a single response and obtain the factuality label from the probe.

[1] We omit residual connection (He et al., 2016) and layer normalization (Ba et al., 2016) for simplicity.

3 Methodology

The construction of PiNose involves three main stages: (1) In the data preparation stage, we bootstrap natural language questions and generate multiple responses, which together serve as model inputs; (2) In the offline consistency checking stage, we employ a peer review mechanism to generate pseudo factuality label for each response; (3) In the probe construction stage, these pseudo factuality labels are used to train a language model probe. Figure 1 illustrates the overall process.

3.1 Stage 1: Data Preparation

In alignment with the requirements of the non-factual content detection task, the supporting data consists of three elements: natural language questions $\mathbf{q}$, their corresponding responses $\mathbf{r}$, and factuality labels f. This stage concentrates on generating large-scale data containing the initial two elements (*i.e.,* $\mathbf{q}$ and $\mathbf{r}$).

Question Bootstrapping. PiNose leverages natural language questions to prompt LLMs to generate responses for consistency checking. However, natural language questions are not always available across all domains. Furthermore, both the diversity and quantity of questions can significantly impact the quality of the prepared data. Therefore, we aim to enable LLMs to bootstrap questions with minimal human involvement.

Fortunately, as Honovich et al. (2023) point out, high-performing language models show significant capacity in question generation. Inspired by these findings, we manually annotate a set of seed questions and employ them as demonstrations for LLMs to generate a large volume of questions via in-context learning (Brown et al., 2020, ICL). To enhance diversity in generation, we broaden the scope of seed questions by incorporating the generated ones and sample diverse combinations from the seed questions for subsequent generation. Detailed prompt for question generation is provided in Figure 4 of Appendix A.5.

Diverse Response Generation. We use previously generated questions as input for LLMs to generate multiple responses for subsequent consistency checking. We design two strategies to encourage the diversity of multiple responses to the same input question. (1) From the perspective of decoding, we adjust the decoding strategy by applying a vanilla sampling method with a relatively high sampling temperature ($t = 1$). (2) From the perspective of model input, we instruct LLMs to answer a question using a variety of prompts (as shown in Figure 5 in Appendix A.5).

The outcome of this stage is a dataset containing questions paired with multiple responses, designated as $\{(\mathbf{q}, \{\mathbf{r}_i\})\}$, where the number of responses $k = |\{\mathbf{r}_i\}|$ serves as a hyperparameter that determines the quantity of responses per question for the subsequent consistency check.

3.2 Stage 2: Offline Consistency Checking

We engage LLMs in the offline consistency check process via a peer review mechanism. First, we gather reviews by asking LLMs to determine for each response whether it is consistent with other responses. Then, we enrich reviews by sampling multiple consistency judgements by varying model inputs. Finally, we integrate reviews to form the pseudo-factuality label for each response and filter out low-quality responses.

Review Gathering. Formally, consistency reviewing involves asking an LLM to evaluate whether the response $\mathbf{r}_i$ to the question $\mathbf{q}$ is semantically consistent with other responses (*i.e.,* $\mathbf{r}_j, j \neq i$). If $\mathbf{r}_i$ has equivalent meaning with other responses, it is considered factual. To ensure unambiguous judgment, we require the LLM to make pairwise comparisons with other $k-1$ responses. For each comparison, it must output one of three labels: "`Consistent`", "`Neutral`", or "`Non-Consistent`". To achieve this, we specify the output format with in-context demonstrations and prompt instructions (as shown in Figure 6 in Appendix A.5).

Review Enrichment. To enhance the diversity of reviews, we introduce variability in the input provided to the LLM during consistency assessments. Recognizing the significant impact of demonstrations on LLM judgments in ICL (Wang et al., 2023b), we utilize a range of diverse demonstration combinations for ICL to elicit varied reviews from the LLM for each pairwise comparison. Diverse demonstrations facilitate the collection of multiple reviews, each potentially providing a unique perspective. In total, we gather N round of reviews for each pairwise comparison, where N is a hyperparameter.

Integration and Filtering. We integrate N reviews for each pairwise comparison, and subsequently integrate $k-1$ pairwise comparisons for each response through the same majority voting mechanism. Here is how the voting works: we first

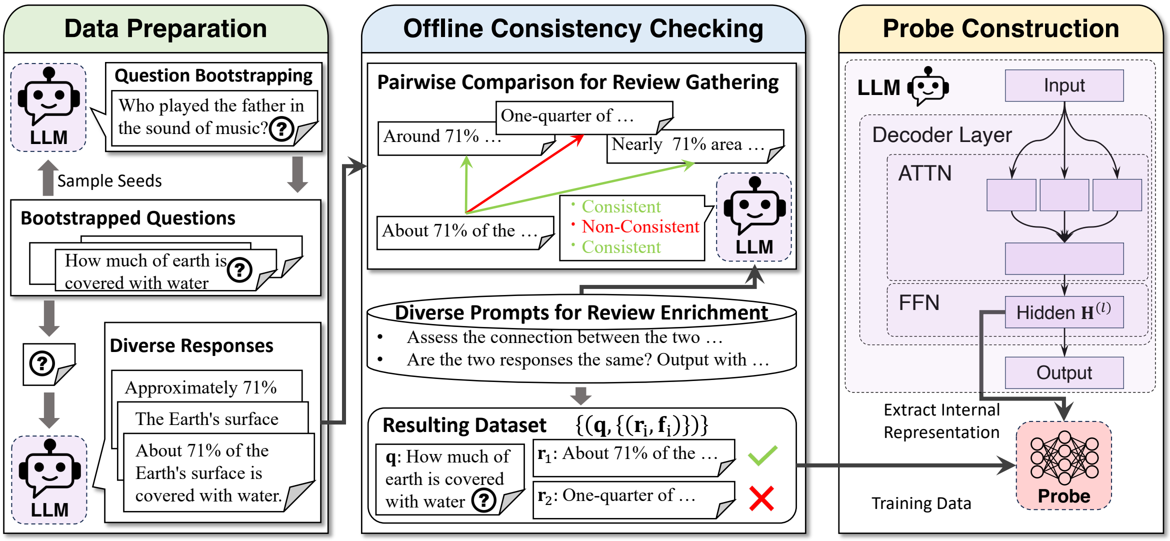

Data Preparation
LLM
Question Bootstrapping
Who played the father in the sound of music?
Sample Seeds
Bootstrapped Questions
How much of earth is covered with water
Diverse Responses
Approximately 71%
The Earth's surface
About 71% of the Earth's surface is covered with water.
LLM
Offline Consistency Checking
Pairwise Comparison for Review Gathering
One-quarter of …
Around 71% …
Nearly 71% area …
About 71% of the …
Consistent
Non-Consistent
Consistent
LLM
Diverse Prompts for Review Enrichment
Assess the connection between the two …
Are the two responses the same? Output with …
Resulting Dataset $\{(\mathbf{q}, \{(\mathbf{r}_i, \mathbf{f}_i)\})\}$
q: How much of earth is covered with water
r1: About 71% of the …
r2: One-quarter of …
Probe Construction
LLM
Input
Decoder Layer
ATTN
FFN
Hidden $\mathbf{H}^{(l)}$
Output
Extract Internal Representation
Training Data
Probe

Figure 1: The overall architecture of PiNose.

consider Neutral consistency judgement as an abstention for voting. Then, to guarantee the quality of the final dataset, we exclude controversial judgements where no single label (Consistent, Neutral, Non-Consistent) receives over 50% of the votes. This step ensures that only the most widely agreed-upon judgements are retained for analysis. Finally, we assign the factuality label True (False) to responses that are predominantly considered consistent (non-consistent) with others.

This stage outputs the dataset with full elements for consistency checking, *i.e.*, $\{(\mathbf{q}, \{(\mathbf{r}_i, f_i)\})\}$.

3.3 Stage 3: Probe Construction

PiNose predicts the factuality of responses via a probing model, as defined in Equation 1. To be more specific, PiNose integrates the response with the question, both formatted according to the template outlined in Figure 7 in Appendix A.5, into the LLM for detection. Subsequently, the hidden representation of the last token in the response at the middle layer of the LLM is employed as the input for the probing model. We train the probing model to maximize the probability of the factuality label while freezing all the parameters of the LLM. Formally, the construction process of the probe optimizes the following cross-entropy loss:

$$\text{loss} = -\sum_{\{\mathbf{q}, \mathbf{r}_j, f_j\}} \log \text{Probe}\Big(\mathbf{H}^{(l)}[i]\Big)\, \mathbb{1}\Big(f_j = \texttt{True}\Big),$$

where $\mathbb{1}(\cdot)$ is the indicator function, i is the index of the last input token, and l represents half the layer number of the LLM to be detected.

3.4 Discussion

We discuss the rationality of PiNose and the involvement of LLMs for implementing PiNose.

The reason of why PiNose successfully detect non-factual responses comes from the model and data perspective. (1) **Model property.** LLMs are well-calibrated after massive pre-training (Kadavath et al., 2022; Zhu et al., 2023). This indicates that for non-factual responses, LLMs tend to assign less probability, while preserve relatively high probability for factual generations. This calibration property guarantees the feasibility of distilling factuality detection dataset from offline consistency checking. It also suggests that the internal states of LLMs tracks whether they are producing factual contents, which PiNose tries to uncover with probing model. (2) **Data quality.** Offline consistency checking gathers diverse instances of inconsistency between responses from LLMs, potentially enhancing the quality of training data for the probing model. Consequently, it enables the model to address a broader range of inconsistency scenarios compared to online consistency checking. Moreover, as the data collection process is fully automated, the dataset can be significantly larger than existing training data for factuality probes. The feasibility of this principle is also widely verified in distant supervision (Quirk and Poon, 2017).

To implement PiNose, LLMs are multiply invoked during the construction process, including data preparation, peer reviewing in consistency checking, and finally non-factual detection. For a coherent implementation, we employ the same

	True-False	NQ	TriviaQA	WebQ
#Train	5,000	N/A	N/A	N/A
#Test	1,000	1,000	1,000	1,000
%True	40.5	46.6	49.5	58.3

Table 1: Data statistics. #Train and #Test are the number of instances in the training data and test set, respectively. %True is the ratio of True labels in the test set. N/A indicates the absence of factuality labels associated with question responses, although the questions in these datasets are present.

LLM for detecting the factuality of responses as the one used for generation and checking consistency. This implementation strategy aligns with our setting, where no third-party LLM is available, and it also enhances the transferability of our method.

4 Experiment

We conduct experiments to examine the performance of PiNose by comparing it to baseline methods for factuality detection. Additionally, we assess its transferability and efficiency.

4.1 Experiment Setup

4.1.1 Datasets

The datasets include both factuality detection benchmark and variations of QA datasets. We introduce the purpose to incorporate each dataset and their data specifications. Detailed statistics are shown in Table 1.

Benchmark. We follow previous research to use **True-False** benchmark (Azaria and Mitchell, 2023b). True-False provides statements generated by LLMs along with corresponding factuality labels examined by humans. It does not include questions for each statement (*i.e.,* $\mathbf{q} = \varnothing$). True-False comes with both training dataset and test dataset.

Variation of QA. Given that probing statements, such as those in True-False dataset, is less practical compared to examining responses to questions from LLMs, for practical evaluation, we establish test sets based on existing QA datasets: **Natural Questions (NQ)** (Kwiatkowski et al., 2019), **TriviaQA** (Joshi et al., 2017), and **WebQ** (Berant et al., 2013). In particular, we sample $1,000$ questions from each dataset and employ Llama2-7B (Touvron et al., 2023b) to generate responses accordingly. The factuality label for each response is annotated by human annotators through comparing with the original ground-truth of each question.

4.1.2 Baselines

We compare PiNose against probing-based and consistency-checking-based methods. Additionally, to ensure a comprehensive comparison, we implement heuristic confidence-based methods as baselines.

- **Probing Based: SAPLMA** (Azaria and Mitchell, 2023a) utilizes a feed-forward neural network for factuality detection, trained on the True-False training data. **RepE** (Zou et al., 2023) conduct principal component analysis (PCA) on the internal representations of True-False training data. It selects a factual direction vector using the factuality labels. During testing, RepE compute the dot product between the internal representation of the given response and the factual direction vector.
- **Consistency Checking Based:** We compare against SelfCheckGPT (Manakul et al., 2023b), which performs factuality detection via online self-consistency checking. We implement two variants. **SelfCheckGPT-NLI** (SCGPT-NLI) uses a BERT-based NLI model (Williams et al., 2018) for consistency checking, while **SelfCheckGPT-Prompt** (SCGPT-PT) elicits LLM itself to evaluate the consistency between two responses via prompt. The prompt used for SCGPT-PT is shown in Figure 8 in the Appendix.
- **Confidence Based:** We also utilize model confidence as an indicator for factuality detection. **Perplexity-AVE** (PPL-AVE) and **Perplexity-Max** (PPL-MAX) (Kadavath et al., 2022; Azaria and Mitchell, 2023a; Zou et al., 2023) quantify the average and maximum token-level probabilities of statements within each test set generated by the evaluated LLM. **It-is-True** (Azaria and Mitchell, 2023a) compares the probabilities between sentences "*It is true that* $\mathbf{q}||\mathbf{r}$." and "*It is false that* $\mathbf{q}||\mathbf{r}$.", where $||$ denotes concatenation.

It is worth noting that probing based baselines rely on training data. We thus implement them using the training dataset from True-False. Besides, as SCGPT-PT and SCGPT-NLI needs input questions to generate multiple responses, it is infeasible to test on True-False, where we mark their results as "N/A" in Table 2.

4.1.3 Evaluation Metrics

We follow conventions (Azaria and Mitchell, 2023b) in factuality detection, employing the area

	True-False		NQ		TriviaQA		WebQ	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
RepE	63.3	61.9	62.6	60.8	67.5	64.3	65.5	63.5
SAPLMA	**90.2**	**83.7**	69.7	67.8	69.3	64.5	75.6	69.5
SCGPT-PT	N/A	N/A	72.3	70.3	76.4	69.9	72.5	72.5
SCGPT-NLI	N/A	N/A	77.4	71.0	76.9	72.1	77.4	72.3
PPL-AVE	67.1	64.1	61.8	59.2	61.7	58.5	63.7	63.3
PPL-MAX	60.7	61.7	54.2	55.0	57.6	57.0	56.4	60.0
It-is-True	54.8	60.1	63.0	59.5	61.7	59.1	75.6	71.0
PiNose	86.6	82.1	**80.4**	**72.5**	**83.9**	**73.8**	**83.3**	**76.0**

Table 2: Overall performance over four test sets.

under the receiver operating characteristic curve (AUC) and accuracy (ACC) as evaluation metrics.

4.1.4 Implementation Details

To implement PiNose, we uniformly use Llama2-7B for data preparation, consistency checking, and factuality detection. For hyperparameters, we set the number of sampled responses N to 9 and the round of peer review k to 7. For fair comparison, we also allow SelfCheckGPT to generate $N = 9$ responses for consistency checking. The training dataset for PiNose consists of $20,000$ constructed triplets $\{(\mathbf{q}, \mathbf{r}_i, f_i)\}$. The threshold for calculating accuracy is determined by selecting the value that yields the highest accuracy among 100 validation instances partitioned from the test sets.

4.2 Main Results

Table 2 presents the AUC and ACC scores for all the compared methods across four test sets. Meanwhile, Table 3 provides insights into the average detection time required by SCGPT-NLI and our PiNose for each instance. **In general, PiNose outperforms probing-based methods across all QA variation dataset, despite being trained without annotated labels. Additionally, PiNose exhibits superior performance compared to consistency checking methods and is also more efficient.** Some detailed findings include:

Leveraging factuality labels substantially improves factuality detection accuracy. This is evidenced by PiNose 's superior performance trained on factuality labels, compared to confidence-based methods such as PPL and It-is-True.

Limitations of annotated labels on model transferability. Despite being trained using annotated labels, probing-based methods like RepE and SAPLMA lag behind a large margin compared to PiNose on the three QA variation test sets. This disparity arises because the two baselines are trained on True-False's training data, which consists of statements rather than question and responses. This difference in input distribution significantly limits the transferability of these models to out-of-distribution datasets. In contrast, PiNose is trained on a diverse range of questions, leading to superior performance across the QA datasets. The slight lag observed in True-False from SAPLMA is also attributed to the model's training on questions. However, probing the factuality of responses to questions is more practical than evaluating statements given by True-False. Therefore, training on datasets guided questions is reasonable.

	NQ	TriviaQA	WebQ
SCGPT-NLI	2.530	2.210	2.050
PiNose	0.024	0.023	0.024

Table 3: Average time required for detecting each instance (in seconds).

Self-consistency correlates well with factuality. Despite SCGPT lacking supervision from factuality labels, it surpasses supervised probing-based baselines, suggesting a strong correlation between its self-consistency principle and factuality. Furthermore, PiNose, also adhering to the self-consistency principle, outperforms SCGPT. This is due to PiNose being exposed to numerous instances with diverse inconsistencies between responses, unlike SCGPT, which focuses solely on responses related to the given question. Moreover, PiNose evaluates the consistency of internal representations rather than discrete output responses like SCGPT, allowing it access to a wider range of information, thereby enhancing its predictive accuracy.

PiNose's detection time is significantly shorter than SCGPT, as shown in Table 3. This is because PiNose relies on offline consistency checking, incorporating consistency characteristics into internal representations during training. As a result, its online inference depends solely on internal representations, eliminating the need for multiple online inferences like those performed by SCGPT.

4.3 Cross-model Evaluation

To implement PiNose, LLMs are invoked multiple times during the construction process, including data preparation, peer reviewing in consistency

#	Data Preparation	Consistency Checking	Factuality detection	AUC	ACC
1	Llama2-7B	Llama2-7B	Llama2-7B	80.4	72.5
2	Llama2-7B	Llama2-7B	Llama2-13B	81.1	73.2
3	Llama2-7B	Llama2-7B	Mistral-7B	81.3	73.1
4	Llama2-7B	Llama2-13B	Llama2-13B	81.7	73.5
5	Llama2-7B	Mistral-7B	Mistral-7B	81.4	73.3

Table 4: Cross-model evaluation performance on NQ. We explore different combinations of LLMs across each of the three stages to evaluate their effectiveness.

checking, and finally non-factual detection. By default, we employ the same LLM for all stages, leveraging an LLM's calibration property. Additionally, we explore whether training data generated by one LLM can effectively train a probe to detect the factuality of content generated by other LLMs. To study this, we use Llama2-7B for data preparation but vary the detection target to Llama2-13B and Mistral-7B. We further switch the LLM for consistency checking to Llama2-13B and Mistral-7B, consistent with the model to be detected. It's important to note that for a given LLM to be detected, the probe needs to be aligned with it, specifically in terms of probe's input, which consists of the internal representation of the response along with the question, that must be generated by the LLM. The crucial findings, as presented in Table 4, include:

More powerful LLMs brings better detection performance. Comparing group 2 and 3 with group 1, where the training data remains consistent (created by Llama2-7B), probes built based on more powerful LLMs demonstrate higher performance, attributed to the enhanced representational capacity of these models.

Generated data facilitates probing across various LLMs. Switching the LLM for consistency checking to match the LLM being detected results in comparable performances between groups 4 and 2, as well as between groups 5 and 3, respectively. This indicates that we can generate the training data, comprising (question, response, factuality label) triplets, once, regardless of the LLMs being probed, and utilize them uniformly to train probes for any LLM.

4.4 Ablation Studies on Data Preparation

We examine the impact of question distribution in the data preparation stage with two variants:

- **PiNose with self questions:** Assumes a scenario where the training dataset consists of questions from the same distribution as the test questions. We utilize questions from the training data that belong to the same dataset as the test set.
- **PiNose with external questions:** Considers a scenario where questions from the same distribution as the test questions are unavailable. We utilize questions from the training data that belong to a different dataset from the test set.

We evaluate these variants across three QA test sets, maintaining a consistent number of training questions ($1,000$ per set) for the first variant. For the second variant, we collect $5,000$ training questions from the remaining two datasets for each target test set. Additionally, we vary the number of generated questions within the range of [1K, 2K, 3K, 4K, 5K, 10K] to assess its impact. We maintain fairness in our evaluation approach by applying an identical labeling methodology, specifically offline consistency checking, to each dataset.

Figure 2(a)(b)(c) presents the performance of these variants on the three QA test sets. We find:

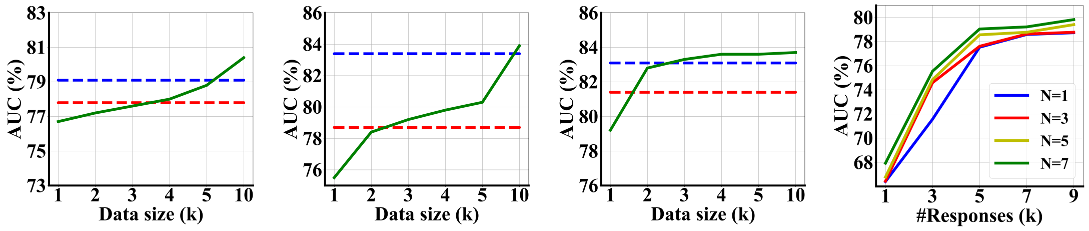
Training on questions of the same distribution as the test set yields significantly better results than a different distribution. Despite having more external questions ($5,000$ vs $1,000$), the second variant still lags behind the first.

Training on generated questions could enhance transferability of the probe. Across the three test sets, training with fewer than $5,000$ generated questions (approximately $3,000+$, $2,000+$, $1,000+$ questions on NQ, TriviaQA, and WebQA, respectively) can achieve performance comparable to using $5,000$ external questions. Additionally, training with approximately $6,000$, $10,000$, and $3,000$ generated questions on these three datasets respectively could outperform training using $1,000$ self-questions. These results indicate that generated questions offer greater diversity, facilitating the transferability of the probe across different test sets, despite the diverse distributions observed among the three test sets.

4.5 Ablation Studies on Consistency Checking

We investigate the impact of two hyperparameters, k (the number of responses) and N (the number of review rounds per response), in the consistency checking stage.

Figure 2(d) displays the detection performance on NQ with varying values of k from 1 to 9 with interval 2 and different values of N (1, 3, 5, 7).



(a) NQ (b) TriviaQA (c) WebQA (d) N&k on NQ

Figure 2: Effects of question generation and the number of reviews and responses. We assess three question distributions for factual detection training data: "**self questions**" ($1,000$ questions from the training data within the same question distribution.), "**external questions**" ($5,000$ questions from a different dataset), and our proposed approach, "**generated questions**" (without relying on available questions). Subfigures (a)-(c) demonstrate the effects of different question distributions on various test sets, while subfigure (d) presents the effects of various k (the number of responses) and N (the round of reviews per response) on NQ.

Model	AUC	ACC
first-layer	52.0	53.3
middle-layer	80.4	72.5
last-layer	76.4	70.3
average token	77.1	70.9
last token	80.4	72.5

Table 5: Evaluating probe construction using internal representations from different layers, either averaged or using the last token.

It's worth noting that $N = 1$ corresponds to the review strategy in SCGPT. Remarkably, the best performance is achieved with $N = 7$, indicating that multiple inferences from an LLM, each guided by different demonstrations acting as instructions, contribute to more robust and confident review outcomes akin to opinions from multiple reviewers. The figure also illustrates that the performance exhibits a smooth increase as more responses are used, also suggesting that multiple responses could result in more confident consistency checking.

4.6 Ablation Studies on Probe Construction

We investigate feature selection at the probe construction stage, exploring the use of internal representations from the last (32nd), middle(16th), and first layers of Llama2-7B. Additionally, we experiment with averaging representations of all tokens within a layer or using only the last token. The default configuration includes the middle-layer representation and the last token in a layer. The results, as depicted in Table 5, indicate that the middle-layer representation and the last token are optimal choices within our setting.

5 Related work

Factuality detection for LLM generated content mainly falls into two categories: consistency-based and probing-based.

Consistency-based methods detect non-factual generations by comparing model generated content with other information. Among these methods, the most widely adopted assumption is that, LLMs usually fail to give consistent responses to the same prompt when generating multiple times (Elazar et al., 2021; Mündler et al., 2023; Pacchiardi et al., 2023), thus motivating a series method to detect non-factual content (Manakul et al., 2023b; Cohen et al., 2023; Azaria and Mitchell, 2023a) or reduce non-factual generations (Dhuliawala et al., 2023; Kuhn et al., 2023). Another thread of works evaluate model self-consistency via model's confidence to the generated content (Kadavath et al., 2022; Azaria and Mitchell, 2023a; Zou et al., 2023). Except for self-consistency checking, there are also attempts that use consistency between model generated content and external information as factuality indicator (Wang et al., 2023c; Gao et al., 2023; Chern et al., 2023).

Probing-based methods possesses the belief that the hidden representation entails certain property of generated content and can be extracted via a light weight model (Alain and Bengio, 2017; Gurnee and Tegmark, 2023). Probing whether LLMs are producing factual content is proved to be feasible (Kadavath et al., 2022), thus motivating researchers to develop more accurate probes (Azaria and Mitchell, 2023a; Zou et al., 2023; Chen et al., 2023). Comparing with these works, which rely on annotated training data, PiNose provides a method that distill consistency patterns from LLMs into a probe.

6 Conclusion

This paper presents PINOSE, a probing method for non-factual content detection that learns from offline consistency checking. PINOSE achieves good transferability among different distributed datasets as its does not rely on manually annotated data. It also avoids the computational burden for online consistency checking. In the future, PINOSE potentially paves way to build more faithful LLMs.

Limitations

The limitations of this work are as follows: (1) **Data Preparation:** PINOSE employs an offline consistency checking method to automatically generate factuality labels for training a probe model. Although the probe model efficiently infers the factuality label of a response from an LLM in a single pass, the offline data preparation stage requires a large amount of data. This involves multiple inferences of the LLM for generating questions, responses, and reviews, resulting in high offline construction costs. Fortunately, as online usage increases, the amortized cost of offline construction decreases. (2) **Open-sourced LLMs:** PINOSE is limited to detecting factuality errors in open-sourced LLMs because it requires the internal representation of the input response along with the question as features input to the probing model for detection. (3) **Factuality Error:** PINOSE is constrained to detecting the factuality errosr in responses to questions or statements. Other aspects of errors, such as logical error, require further investigation.

Ethical Considerations

We discuss the ethical considerations and broader impact of this work in this section: (1) **Intellectual Property:** The datasets employed in this study, comprising True-False, NQ, TriviaQA, and WebQ, are widely accessible and established resources designed to facilitate extensive research in artificial intelligence and natural language processing (NLP). We are confident that these resources have been adequately de-identified and anonymized. (2) **Data Annotation:** We recruit 10 annotators from commercial data production companies to label factual accuracy for three QA test sets, seed questions for question generation, and consistent/inconsistent response pairs for consistency checking. Annotators are compensated fairly based on agreed-upon working hours and rates. Prior to annotation, annotators are briefed on the data's processing and usage, which is formalized in the data production contract. (3) **Intended Use:** The proposed PINOSE is utilized for detecting non-factual content generated by LLMs. (4) **Misuse Risks:** There is a risk that PINOSE could be exploited for adversarial learning, potentially enabling LLMs to generate more implicit non-factual content that is more challenging to detect. (5) **Potential Risk Control:** The trained PINOSE is made publicly available to the open-source community, which may help mitigate the risks associated with its potential misuse for adversarial learning.

Acknowledgments

This work is supported by the National Key Research & Develop Plan (2023YFF0725100), the National Natural Science Foundation of China (62322214), Zhipu Project Fund, and a grant from the Institute for Guo Qiang, Tsinghua University (2019GQB0003).

References

Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2023. Exploring the state of the art in legal qa systems. *Journal of Big Data*, 10(1).

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Anonymous. 2023. INSIDE: LLMs' internal states retain the power of hallucination detection. In *Submitted to The Twelfth International Conference on Learning Representations*. Under review.

Amos Azaria and Tom Mitchell. 2023a. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Amos Azaria and Tom Mitchell. 2023b. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *ArXiv preprint*, abs/1607.06450.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, pages 207–219.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 245–255.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *ArXiv preprint*, abs/2307.13528.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *ArXiv preprint*, abs/2309.11495.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *ArXiv preprint*, abs/2310.02207.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for open-domain qa. *ArXiv preprint*, abs/2212.08635.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023a. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023b. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *ArXiv preprint*, abs/2305.15852.

Lorenzo Pacchiardi, Alex J Chan, Sören Mindermann, Ilan Moscovitz, Alexa Y Pan, Yarin Gal, Owain Evans, and Jan Brauner. 2023. How to catch an ai liar: Lie detection in black-box llms by asking unrelated questions. *ArXiv preprint*, abs/2309.15840.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2023. Recitation-augmented language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *ArXiv preprint*, abs/2310.07521.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023b. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *ArXiv preprint*, abs/2203.11171.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, et al. 2023c. Factcheck-gpt: End-to-end fine-grained document-level fact-checking and correction of llm output. *ArXiv preprint*, abs/2311.09000.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *ArXiv preprint*, abs/2209.10063.

Chiwei Zhu, Benfeng Xu, Quan Wang, Yongdong Zhang, and Zhendong Mao. 2023. On the calibration of large language models and alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *ArXiv preprint*, abs/2310.01405.

A Appendix

A.1 Perplexity-based Baseline

The perplexity-based baseline is formulated as:

$$\text{PPL(AVE)} = -\frac{1}{J}\sum_{j}\log p_{ij}, \qquad (2)$$

$$\text{PPL(MAX)} = \max_{j}(-\log p_{ij}), \qquad (3)$$

where i denotes the i-th statement, and j denotes the j-th token in the i-th statement. J is the number of tokens in the i-th statement. p_{ij} denotes the probability of the j-th token in the i-th statement generated by the LLM. PPL(AVE) measures the average likelihood of all tokens, while PPL(MAX) measures the likelihood of the least likely token in the statement.

#	Data Preparation	Consistency Checking	Factuality detection	AUC	ACC
1	Llama2-7B	Llama2-7B	Llama2-7B	83.9	73.8
2	Llama2-7B	Llama2-7B	Llama2-13B	83.8	77.0
3	Llama2-7B	Llama2-7B	mistral-7B	86.8	76.8
4	Llama2-7B	Llama2-13B	Llama2-13B	83.7	77.1
5	Llama2-7B	mistral-7B	mistral-7B	86.3	76.9

Table 6: Cross-model evaluation performance on TriviaQA.

#	Data Preparation	Consistency Checking	Factuality detection	AUC	ACC
1	Llama2-7B	Llama2-7B	Llama2-7B	83.3	76.0
2	Llama2-7B	Llama2-7B	Llama2-13B	83.4	76.3
3	Llama2-7B	Llama2-7B	mistral-7B	83.1	76.5
4	Llama2-7B	Llama2-13B	Llama2-13B	83.4	76.4
5	Llama2-7B	mistral-7B	mistral-7B	83.1	76.6

Table 7: Cross-model evaluation performance on WebQ.

A.2 Cross-model Evaluation on TriviaQA and WebQ

We present the cross-model evaluation results on TriviaQA in Table 6 and on WebQ in Table 7 to explore whether training data generated by one LLM can effectively train a probe to detect the factuality of other LLMs. The settings are the same as those presented for NQ in Table 4. In addition to the default setting where we employ the same LLM for all stages of data preparation, consistency checking, and factuality detection, we also investigate two other settings. The first setting involves using Llama2-7B for data preparation but varying the detection target to Llama2-13B and Mistral-7B. The second setting involves further switching the LLM for consistency checking to Llama2-13B and Mistral-7B, consistent with the LLM to be detected.

Tables 6 and 7 present observations: for the first setting, when the detection target is changed to more powerful LLMs, the detection performance increases, indicating that more powerful LLMs can improve detection performance. Additionally, for the second setting, when changing the LLM for

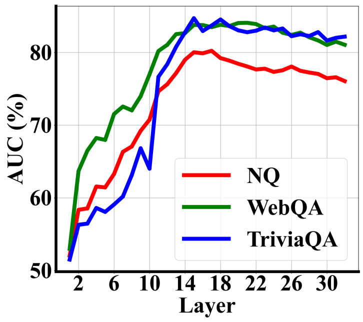


Figure 3: AUC obtained using the internal representations of different layers at the probe construction stage.

consistency checking to the same more powerful LLMs, the detection performance remains almost unchanged. This suggests that we can generate training data for questions, responses, and reviews once, regardless of the LLMs being probed, and uniformly employ them to train probes for any LLMs.

A.3 Evaluation of Different Layers

We vary the internal representations obtained per layer from the 1st to the last layer (32nd) of Llama2-7B, use them to construct probes respectively, and show the evaluated AUC on the three QA test sets in Figure 3. The results demonstrate that the detection performance of PiNose generally increases and then decreases with the increase in the number of layers. Typically, the best performance is achieved at the middle layer.

A.4 Training Dataset Samples

As shown in Table 8, the extensive internal knowledge of Large Language Models (LLMs) enables them to generate datasets encompassing a broad array of topics. This diversity enhances the effectiveness of training downstream detectors.

A.5 Employed Prompts

We introduce the prompts used at different steps of our proposed PiNose, and also other baseline methods that need prompts.

Prompt for Question Generation in PiNose is shown in Figure 4, where the seed questions are randomly sampled from an initial set of questions annotated by humans, which is then expanded by the newly generated questions.

Prompt for Response Generation in PiNose is shown in Figure 5, where five instructions for generating responses are presented. We randomly select an instruction from this set each time to encourage diverse response generation.

Prompt for Consistency Checking in PiNose is depicted in Figure 6. It instructs LLMs to determine whether two responses are "`Consistent`", "`Neutral`", or "`Non-Consistent`" given a question. For each judgment, three demonstrations are randomly selected from a set of 16 consistency judgment pairs. These diverse demonstrations facilitate the collection of multiple judgments (reviews) for each comparison between two responses, potentially offering unique perspectives.

Prompt for Probe Construction in PiNose is depicted in Figure 7, where a question and the answer generated by the LLM under detection are organized as input for the LLM to obtain their internal representation. This representation then serves as input for the probe to predict its factual label.

Prompt for SelfCheckGPT-Prompt is displayed in Figure 8, identical to the one presented in (Manakul et al., 2023b). SelfCheckGPT-Prompt facilitates consistency checking by presenting a sentence and its context to LLM, enabling it to judge whether the context adequately supports the sentence.

Table 8: Sample of LLM Generated Training Dataset

Question	**Response**	**Label**
what composer wrote the music for the lord of the rings film trilogy?	The music for the Lord of the Rings film trilogy was composed by Howard Shore.	True
when did charles dickens publish "a christmas carol"?	Charles Dickens published "A Christmas Carol" in 1843.	True
who sings the song with every beat of my heart?	Lindsay Lohan sings with every beat of my heart.	False
what was the peak unemployment rate during the us great depression?	47.7% in 1933.	False
what is the latest operating system (os) for apple watch?	The latest operating system for Apple Watch is watchOS 7.	False

Prompt for Question Generation in PiNose.

Please ask some objective questions of similar difficulty to [Seed Questions].

### [Seed Questions]
1. which part of earth is covered with water?
2. what is the military equivalent of a gs-14?
3. who provided the voice for the geico insurance company gecko?
4. who played the father in sound of music?
5. fugees killing me softly with his song original?
6.

Figure 4: Prompt for question generation in PiNose. Five seed questions from NQ are provided and the blank following item 6 is the new question that encourages LLMs to generate.

Prompt 1 for Response Generation in PiNose.

```
### Question
where is taurus the bull in the night sky
### Answer
```

Prompt 2 for Response Generation in PiNose.

```
### Instruction
Answer the following question.


### Question
where is taurus the bull in the night sky
### Answer
```

Prompt 3 for Response Generation in PiNose.

```
### Instruction
Give a helpful answer.


### Question
where is taurus the bull in the night sky
### Answer
```

Prompt 4 for Response Generation in PiNose.

```
### Instruction
Generate a brief response in just one sentence.


### Question
where is taurus the bull in the night sky
### Answer
```

Prompt 5 for Response Generation in PiNose.

```
### Instruction
Compose a concise answer within a single sentence.


### Question
where is taurus the bull in the night sky
### Answer
```

Figure 5: Prompt for response generation in PiNose. Five different instructions are randomly employed to elicit diverse responses.

Prompt for Consistency Checking in PINOSE.

```
Assess the connection between the two responses to the initial query, taking into account the potential scenarios of Endorsement, Contradiction, and Impartiality.

### Input

- **Question:** where is vina del mar located in chile
- **First Response:** Vina del Mar is located in the Valparaíso Region of Chile, approximately 120 kilometers west of Santiago.
- **Second Response:** Viña del Mar is a city located on the central coast of Chile. It is part of the Valparaíso Region and is situated about 120 kilometers (75 miles) northwest of Santiago, the capital of Chile.

### Output
Judgement: Endorsement
@Reason@: The two responses provide consistent information about the location of Viña del Mar in the Valparaíso Region of Chile, approximately 120 kilometers west/northwest of Santiago. The details in both responses align, endorsing the accuracy of the information.


### Input

- **Question:** where is taurus the bull in the night sky
- **First Response:** Taurus the Bull is located in the southeastern part of the sky, near the constellation Orion and the celestial equator.
- **Second Response:** Taurus the Bull is located in the eastern part of the night sky, stretching from the constellation Orion to the constellation Gemini.

### Output
Judgement: Contradiction
@Reason@: The two responses provide different information regarding the location of Taurus the Bull in the night sky.


### Input

- **Question:** who is the old man in waiting on a woman
- **First Response:** The old man in the waiting room is Mr. Johnson's father.
- **Second Response:** The old man in the picture is likely the grandfather or great-grandfather of the woman he is waiting on, as he appears to be elderly and has a gentle expression on his face.

### Output
Judgement: Impartiality
@Reason@: There is no explicit contradiction between the two responses, and they may collectively provide a more detailed and comprehensive answer to the question. The overall tone is impartial, as the information in the responses is neither conflicting nor mutually supportive.


### Input

- **Question:** where is vina del mar located in chile
 - **First Response:** Vina del Mar is located in the Valparaíso Region of Chile, approximately 120 kilometers west of Santiago.
 - **Second Response:** Viña del Mar is a city located on the central coast of Chile. It is part of the Valparaíso Region and is situated about 120 kilometers (75 miles) northwest of Santiago, the capital of Chile.

### Output
Judgement:
```

Figure 6: Prompt for Consistency Checking in PINOSE. Three demonstrations illustrate judgments of endorsement, contradiction, and impartiality, respectively.

Prompt for Probe Construction in PiNose.

### Instruction
Compose a concise answer within a single sentence.

### Question
where is taurus the bull in the night sky

### Answer
Taurus the Bull is located in the southeastern part of the sky, near the constellation Orion and the celestial equator.

Figure 7: Prompt for Probe Construction in PiNose. The LLM under detection receives the question and its generated answer to obtain the corresponding internal representation. This representation serves as the input for training the probe.

Prompt for SelfCheckGPT with Prompt.

Context: Delhi was made the capital of India for the first time by the British East India Company in 1858, when the British assume control of the Indian subcontinent following the Indian Rebellion of 1857.

Sentence: Delhi was first made the capital of India by the Mughal emperor Shah Jahan in the 17th century.

Is the sentence supported by the context above? Answer Yes or No.

Answer:

Figure 8: Prompt for SelfCheckGPT-Prompt. A sentence and its context are provided to enable LLMs to determine whether the sentence is supported by the context.