LLM Factoscope: Uncovering LLMs' Factual Discernment through Measuring Inner States

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have revolutionized various domains with extensive knowledge and creative capabilities. However, a critical issue with LLMs is their tendency to produce outputs that diverge from factual reality. This phenomenon is particularly concerning in sensitive applications such as medical consultation and legal advice, where accuracy is paramount. Inspired by human lie detectors using physiological responses, we introduce the LLM Factoscope, a flexible and extendable pipeline that leverages the inner states of LLMs for factual detection. Our investigation reveals distinguishable patterns in LLMs' inner states when generating factual versus non-factual content. We demonstrate its effectiveness across various architectures, achieving over 96% accuracy on our custom-collected factual detection dataset. Our work opens a new avenue for utilizing LLMs' inner states for factual detection and encourages further exploration into LLMs' inner workings for enhanced reliability and transparency.

1 Introduction

Large Language Models (LLMs) have gained immense popularity, revolutionizing various domains with their remarkable creative capabilities and vast knowledge repositories. These models reshape fields like natural language processing (Khurana et al., 2023), content generation (Acharya et al., 2023), and more. However, despite their advanced abilities, a growing concern surrounds their propensity for "hallucination" — the generation of outputs that deviate from factual reality (Zhang et al., 2023b). In critical applications like medical consultation (Zhang et al., 2023a), legal advice (Cui et al., 2023), and educational tutoring (Upadhyay et al., 2023), factual LLM outputs are not just desirable but essential, as non-factual outputs from these models could potentially lead to negative outcomes for users, affecting their health, legal standing, or educational understanding. Recognizing this, LLM-generated content's factual detection has emerged as an area of paramount importance (Mosca et al., 2023). Current research predominantly relies on cross-referencing LLM outputs with external databases (Zhang et al., 2023b). While effective, this approach necessitates extensive external knowledge bases and sophisticated cross-referencing algorithms, introducing more complexity and dependency. This raises a compelling question: Could we possibly exclude external resources but only leverage the inner states of LLMs for factual detection?

Inspired by human lie detectors, we explore establishing a lie detector for LLMs. Humans show specific physiological changes when making statements that contradict their knowledge or beliefs, enabling lie detection through these changes. We assume that exposure to a wide range of world knowledge during LLMs' training establishes their knowledge or beliefs. Moreover, training datasets usually have more factual than non-factual sources (Touvron et al., 2023). This assumption makes establishing a lie detector for LLMs possible. To achieve this, we need to address two essential questions: (1) What might the "physiological" indicators for LLMs be, similar to heart rate and eye movements in humans? (2) How can we leverage these indicators in LLMs to establish the lie detector? However, the lack of interpretability in LLMs makes it impossible for humans to understand model parameters or hidden states, rendering the solutions to these questions challenging.

We address the above two challenges through heuristic and data-driven methods, respectively. For the first challenge, we initially use *activation maps* as static features to represent the "physiological" states of the LLM. Based on our observations, factual-related activations show higher intensity when the LLM produces factual outputs, making the activation maps representative to serve as fea-

tures for the lie detector. Subsequently, we use *final output ranks*, *top-k output indices*, and *top-k output probabilities* across layers as dynamic features to represent the LLM's decision-making processes. We note that factual outputs by LLMs show more stable final output ranks, greater semantic similarity in top-k output indices, and larger top-k output probability differences in later layers (See Section 3). For the second challenge, due to the intricate relationship between the features and the factuality of outputs, we employ a data-driven approach to extract underlying principles from these features. Consequently, we design a systematic and automated factual detection pipeline, LLM Factoscope, that includes the factual datasets collection, the inner states collection, and the design of the factual detection model (See Section 4).

In our experiments, we empirically demonstrate the effectiveness of the LLM Factoscope across various LLM architectures, including GPT2-XL-1.5B, Llama2-7B, Vicuna-7B, Stablelm-7B, Llama2-13B, and Vicuna-13B. The LLM Factoscope achieves an accuracy rate exceeding 96% in factual detection. Additionally, we extensively examine the model's generalization capabilities and conduct ablation studies to understand the impact of different components and parameters on the LLM Factoscope's performance. Our work paves a new path for utilizing inner states from LLMs for factual detection, sparking further exploration and analysis of LLMs' inner states for enhanced model understanding and reliability.

Our contributions are as follows: (1) We identify effective inner states closely associated with the factual accuracy of the content generated by LLMs, discovered through observation. (2) We introduce the LLM Factoscope, a novel and extendable pipeline for automating the detection of factual accuracy in LLM outputs, facilitating the incorporation of new datasets with ease. (3) Our model has been empirically validated across various LLM architectures, demonstrating over 96% accuracy on our custom-collected factual detection dataset. Datasets and code are released for further research: https://anonymous.4open.science/r/llm_factoscope.

2 Related Work

2.1 Large Language Models

LLMs, based on transformer decoder architecture, are highly capable of capturing complex language patterns with their billions of parameters. These models function as follows: Given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, an LLM maps it to an output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_m)$. Each input token x_i is embedded into a high-dimensional space, forming a sequence of embeddings $\mathbf{E} = \text{Embed}(\mathbf{x})$. The core of an LLM is multiple transformer blocks, each featuring a self-attention module $\mathcal{A}$ and a multilayer perceptron (MLP) $\mathcal{M}$. For each layer l, the process involves the self-attention mechanism acting on the hidden state $\mathbf{H}^{(l-1)}$ (initially $\mathbf{H}^{(0)} = \mathbf{E}$) to produce $\mathbf{A}^{(l)}$, which is then processed by the MLP to yield $\mathbf{M}^{(l)}$. The hidden state for the next layer is calculated as $\mathbf{H}^{(l)} = \mathbf{H}^{(l-1)} + \mathbf{A}^{(l)} + \mathbf{M}^{(l)}$. Finally, after the last layer L, the output $\mathbf{H}^{(L)}$ is converted into a probability distribution over the vocabulary through a linear layer and a softmax function: $\mathbf{y} = \text{softmax}(W \cdot \mathbf{H}^{(L)} + b)$, where W and b are the weights and bias.

2.2 LLM Factual Detection

Fact-checking LLM outputs has become an increasingly critical task. Current approaches to mitigate LLM-generated inaccuracies include scrutinizing training datasets and cross-referencing external databases. Manual examination (Penedo et al., 2023) of training datasets is labor-intensive, while external database referencing (Ren et al., 2023) (Mialon et al., 2023) incurs additional computational costs and relies heavily on the effectiveness of cross-verification techniques. A recently proposed SAPLMA (Azaria and Mitchell, 2023) investigates whether LLMs can discern the factuality of an input sentence. Our method aims to distinguish each output as factual or non-factual, closely simulating the typical situations most users encounter when utilizing these models. We leverage not just activations from a single layer, but also the inter-layer changes in activations and hidden states within the LLM.

3 Observations

In this section, we present an analysis from both static and dynamic perspectives. Statically, we examine activation maps, while dynamically, we project the hidden states in each layer to the probability distribution over the vocabulary humans understand. At the static level, Figure 1(a) demonstrates that when the LLM provides factual information, the factual-related neurons activate more intensely. This could be due to the strengthening

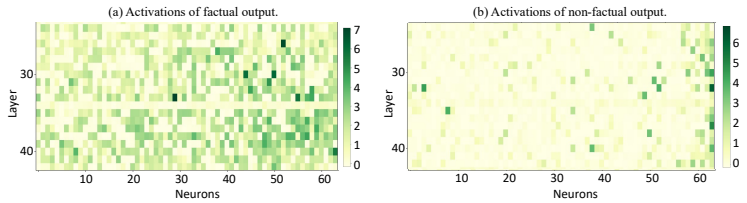

Figure 1: Activation maps of layers 24-42. The figure shows the top 1% neurons that most frequently exhibit the highest activations when the LLM outputs factual.

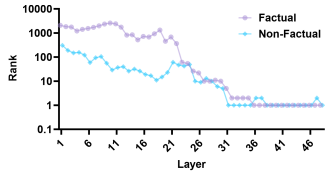

Figure 2: Case study of final output rank.

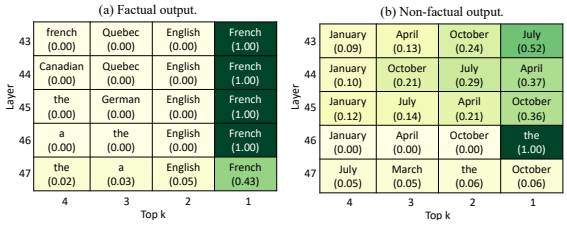

Figure 3: Case study of the top-4 output indices and their probabilities in the last five layers

of these connections from training on factual data, similar to how human memory is reinforced by repeated learning. On the other hand, Figure 1(b) shows a case of non-factual information with less activation in factual-related neurons. This suggests that activation maps potentially serve as indicators for factual accuracy assessment.

From the dynamic perspective, we observe the evolution of the final output ranks across layers, depicted in Figure 2. For instance, when the LLM is queried about the director of "The Shining" and initially outputs "Stanley," we observe its rank across layers, where rank 1 represents the highest probability. In factual outputs, the final output rank ascends to a higher position after layer 22, showing the LLM's growing confidence to finalize its output in the later stages of the network. In contrast, non-factual outputs show rank fluctuations after layer 35, reflecting the model's uncertainty. Further, we observe the top-k output indices and their probabilities. For factual responses, as shown in Figure 3(a), the model consistently prioritizes the same word as the highest probability output. On the other hand, while generating non-factual responses, the top probability words are different in the last few layers, as seen in Figure 3(b), indicating a lack of confidence. Additionally, the semantic similarity between the top-1 output from one layer to the next is greater for factual than non-factual outputs. These observations reveal that changes in final output rank, top-k output indices, and top-k output probabilities between layers potentially serve as indicators for factual detection.

4 LLM Factoscope

In this section, we begin with an overview of our pipeline. Then, we introduce the data collection for factual detection and preprocessing steps for effective model training. Lastly, we present the model architecture of the LLM Factoscope.

4.1 Overview

We propose the LLM Factoscope, a pipeline designed to leverage the inner states of LLMs for factual detection, as depicted in Figure 4. This pipeline facilitates the easy expansion of the factual detection model's dataset coverage. Addressing the absence of word-level factual detection methods, we develop our own dataset by extracting structured factual data from Kaggle (Sculley, 2023). This dataset is then deployed to probe LLMs to check whether their responses align with factual correctness, serving as labels for our inner states dataset. Concurrently, we capture the LLMs' inner states, which include the model's inner representation of knowledge, and use them as features for our dataset. Following data collection and preprocessing, we train a Siamese network-based model designed to minimize the embedding distance between similar class data and maximize the distance between pairs of dissimilar class responses. In Figure 5, we illustrate a practical application that enables users to verify the factuality of the LLM's outputs. Consider a scenario where a user queries an LLM to craft a script for a film introduction video. The user inputs a prompt: "The film titled The Shining was directed by". The LLM responds with "Stanley Kubrick and was released in 1980". The LLM Factoscope marks the "Stanley Kubrick" and the release year "1980" as factual.

4.2 Factual Data Collection

We start our dataset collection by searching for factual-related CSV datasets on Kaggle (Sculley, 2023). The CSV format's inherent structuring into entities, relations, and targets makes it an ideal can-

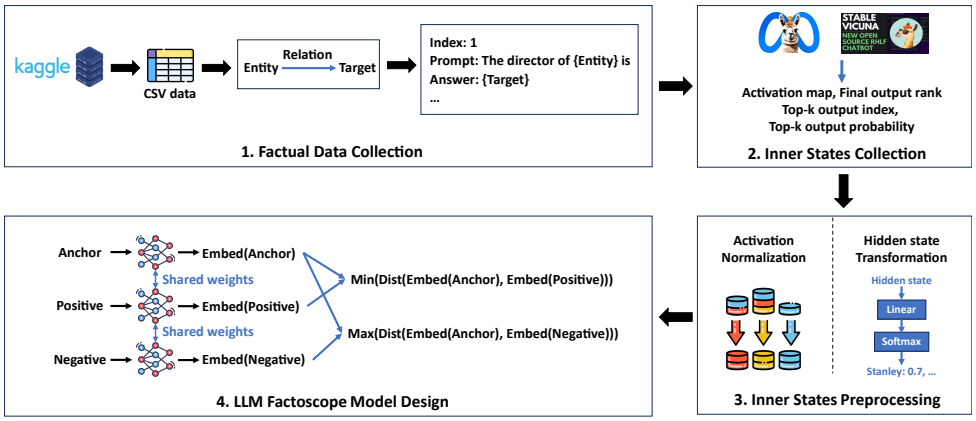

Figure 4: Pipeline of the LLM Factoscope.

Category	Example		Size
Art (Mexwell, 2023)(Grijalvas, 2023)	P: The artist of the artwork Still Life with Flowers and a Watch is	A: Abraham Mignon	67,302
Sport (Guardian, 2023)(Upadhyay, 2023)	P: The athlete Ole Jacob Bangstad represents the country of	A: Norway	31,718
Literary (Naren, 2023)	P: The book Twilight was written by	A: Stephenie Meyer	54,301
Geography (Wikidata, 2023)	P: The city Leipzig is located in the country of	A: Germany	1,103
History (MIT, 2023)	P: The birthplace country of the historical figure Albert Einstein is	A: Germany	56,705
Science (Foundation, 2023)	P: The Nobel laureate Jacobus Henricus van 't Hoff is from	A: Netherlands	8,971
Economics (Elgiriyewithana, 2023)	P: Microsoft was started by	A: Bill Gates	5,228
Multi (Meng et al., 2022)	P: The mother tongue of Danielle Darrieux is	A: French	21,918
Total			247,246

Table 1: Overview of the Factual Dataset. Abbreviation: P-Prompt, A-Answer.

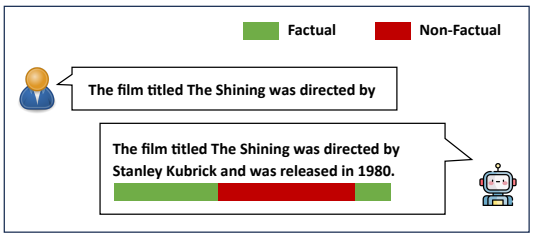

Figure 5: An instance of using LLM Factoscope.

didate for the automated generation of prompts and answers. Our dataset includes various categories and each category includes multiple relations—for instance, in the art category, relationships such as artwork-artist, movie-director, movie-writer, and movie-year are included. We have manually crafted clear prompts to ensure that LLMs can accurately comprehend the questions, guiding them to directly provide answers, thereby avoiding the repetition of the question. To further enhance the dataset's diversity, we have developed multiple synonymous question templates for each relation. Table 1 provides an overview of the datasets. Users can adopt this collection method with other open datasets to easily expand the coverage of the factual dataset. We include a neuron editing dataset from Meng et al. (Meng et al., 2022) to expand the data's scope.

4.3 Inner States Collection

After constructing the factual dataset, we feed the prompts into the LLM, capturing the responses and the inner states associated with the last token of each prompt. In the following, we detail four key types of inner states: activation maps, final output ranks, top-k output indices and probabilities. The first shows the locations and intensity of high activations, representing the static view, while the latter three display the evolution of decision-making and probabilities with a dynamic perspective.

Activation maps: The activation maps represent the activation values of the last token of the prompt when processed by the LLM. These maps contain the LLM's inner representation of the knowledge related to the prompt. As the LLM traverses through its layers, it retrieves information relevant to the prompt (Meng et al., 2023). When the subsequent word aligns with the factual answer, it indicates successful knowledge retrieval at the intermediate layers; otherwise, it suggests inadequate knowledge retrieval. These contrasting scenarios are expected to show distinct activation maps.

Final output ranks: The final output ranks represent the position of the final output token in the

probability distribution at each layer of the LLM. Specifically, we acquire the hidden states at each layer, apply the same vocabulary mapping as done in the final hidden layer through the linear layer, and thereby attain logits for each token in the vocabulary (Vaswani et al., 2017). The rank is determined based on the descending order of logits for the final output token at each layer. The ranks show how the likelihood of the final token changes across layers, reflecting the model's evolving output preferences.

Top-k output indices and probabilities: From the logits used in the final output ranks, we identify the top-k tokens with the highest logits in each layer. These tokens represent the model's most likely outputs after processing the information at each layer. The relationships among these top-k tokens, both within and across layers, shed light on various cognitive aspects of the model's processing. Applying a softmax function to the logits in each layer, we get the probability distribution of all tokens, subsequently identifying the top-k tokens with the highest probabilities. This data reflects the fluctuating probabilities of these tokens across layers, providing insights into the model's probabilistic reasoning.

Alongside these inner states, we record labels for factual detection. These labels, derived by evaluating whether the model's next word following each prompt aligns with the factual answer, serve as a key indicator of the accuracy in factual detection. A correct alignment is marked as positive, while a misalignment is categorized as negative.

4.4 Inner States Preprocessing

In this section, we introduce the preprocessing of inner states for effective integration into the training process. We detail the preprocessing methods applied to each category of inner states.

Normalization of activation map: We calculate the mean μ and standard deviation σ of the dataset. The activation map $\mathbf{A}$ is then normalized using the formula: $\mathbf{A}_{\text{normalized}} = (\mathbf{A} - \mu)/\sigma$. This normalization ensures a uniform scale for the activation values, enhancing their comparability and relevance in the model's learning mechanisms.

Transformation of final output ranks: We adjust the final output ranks to a 0-1 range, highlighting lower values. Mathematically, the transformation of rank r can be represented as $r_{\text{transformed}} = 1/[(1-r) + 1 + 10^{-7}], r \in (1, |V|)$, where $|V|$ is the size of LLM's vocabulary. When the rank r is 1 (indicating the highest rank), the transformed rank $r_{\text{transformed}}$ becomes its maximum value, close to 1. Adding 10^{-7} in the denominator is a small constant to prevent division by zero.

Distance calculation for top-k output indices: In processing the top-k output indices, we measure the semantic similarity across adjacent layers by calculating the cosine similarity between the embeddings of tokens, providing insights into how the model's decision-making and semantic continuity evolves across layers.

It is important to note that while the above inner states require preprocessing to standardize their scales or enhance their interpretability, the top-k output probabilities do not need such preprocessing. This is because the top-k output probabilities are inherently on a consistent scale, being probabilities that naturally range from 0 to 1.

4.5 LLM Factoscope Model Design

After preparing the dataset of inner states, we develop the LLM Factoscope model, inspired by the principles of few-shot learning and Siamese networks. It is designed to effectively learn robust representations from limited data. This approach distinguishes between factual and non-factual content and demonstrates impressive generalization capabilities on similar but unseen data. Our model comprises four distinct sub-models, each dedicated to processing one of the inner states.

For the activation maps, top-k output indices, and top-k output probabilities, we utilize Convolutional Neural Networks (CNNs) with the ResNet18 architecture (He et al., 2016). The choice of ResNet18, with its convolutional and residual connections, is particularly advantageous for efficiently capturing the relationships between and within different layers of the LLM. These CNNs transform the inner states into embeddings $\mathbf{E}_{\text{activation}}$, $\mathbf{E}_{\text{top-k index}}$, and $\mathbf{E}_{\text{top-k prob}}$. Each embedding captures unique aspects of the LLM's processing dynamics. As for the final output ranks, a sequential data type, we use a Gated Recurrent Unit (GRU) network (Cho et al., 2014), reflecting the temporal evolution of the model's output preferences across layers. This network yields an embedding $\mathbf{E}_{\text{rank}}$. The embeddings from these four sub-models are then integrated through a linear layer to form a comprehensive mixed representation, $\mathbf{E}_{\text{mixed}}$, which captures an integrated expres-

sion of the LLM's factual understanding, representing spatial and temporal insights.

During training, our model utilizes the triplet margin loss (Schroff et al., 2015), a metric integral to embedding learning in few-shot learning scenarios. This loss function is designed to minimize the distance between instances of the same class while maximizing the distance between instances of different classes. For a given training instance x, we feed it to the LLM Factoscope model and get an embedding for its mixed representation, $\mathbf{E}_{\text{anchor}}$. Then, we select a positive example $\mathbf{x}_{\text{pos}}$ from the same category as the anchor and a negative example $\mathbf{x}_{\text{neg}}$ from a different category. Subsequently, we obtain their respective mixed expressions $\mathbf{E}_{\text{pos}}$ and $\mathbf{E}_{\text{neg}}$. The triplet margin loss aims to ensure that the distance between the anchor and the positive instance, $\text{Dist}(\mathbf{E}_{\text{anchor}}, \mathbf{E}_{\text{pos}})$, is smaller than the distance between the anchor and the negative instance, $\text{Dist}(\mathbf{E}_{\text{anchor}}, \mathbf{E}_{\text{neg}})$, by at least a margin α. This loss function is formally defined as: $L = \max(\text{Dist}(\mathbf{E}_{\text{anchor}}, \mathbf{E}_{\text{pos}}) - \text{Dist}(\mathbf{E}_{\text{anchor}}, \mathbf{E}_{\text{neg}}) + \alpha, 0)$, where $\text{Dist}(\cdot, \cdot)$ is the chosen distance metric, typically Euclidean distance. By fine-tuning α, we can enhance the model's discriminative capability, ensuring that the distance between the anchor and the positive instance is less than that between the anchor and the negative instance by at least the margin α. The training process minimizes the loss, refining the model's ability to accurately differentiate between factual and non-factual content.

In the testing phase, we establish a support set consisting of data samples and their corresponding targets, denoted as $\{S_1, \ldots, S_n\}$ and $\{T_{\text{sup}_1}, \ldots, T_{\text{sup}_n}\}$, respectively. These samples have not been used in the training process of the LLM Factoscope model. They provide a reference for comparing and classifying new, unseen test data. Each sample in the support set is processed through the LLM Factoscope model to generate mixed representations, represented by $\{\mathbf{E}_{\text{sup}_1}, \ldots, \mathbf{E}_{\text{sup}_n}\}$. The mixed representations are outputs of the LLM Factoscope model. The test data's mixed representation, $\mathbf{E}_{\text{test}}$, is then compared against these support set representations. The classification of the test data is determined by identifying the closest support set embedding to $\mathbf{E}_{\text{test}}$. The target of the test data is the target of this nearest support set data: $T_{\text{test}} = T_{\text{sup}_{i^*}}$ where $i^* = \text{argmin}_i \, \text{Dist}(\mathbf{E}_{\text{test}}, \mathbf{E}_{\text{sup}_i})$. Here, the index i^* identifies the support set data that is closest to the test data, and $T_{\text{sup}_{i^*}}$ is the target associated with this closest support set data. This approach ensures accurate and reliable classification of the test data by leveraging the similarities within the representations of the support set.

5 Evaluation

5.1 Experimental Setup

Dataset. We employ various factual datasets encompassing art, sports, literature, geography, history, science, and economics, comprising 247,246 data points. Then, we record the inner states of the LLM as it provides factual and non-factual outputs, including activation maps, final output ranks, top-k output indices, and top-k output probabilities. The label assigned to each data point indicates whether the corresponding model output is factual or non-factual. To ensure dataset balance, we randomly select an equal number of factual and non-factual data points for each factual relation. The dataset is preprocessed to ensure it is standardized and optimized for model learning.

Models. Our experiments are conducted on several popular LLMs, each with distinctive architectures and characteristics. These models include GPT2-XL (Radford et al., 2019), Llama-2-7B (Touvron et al., 2023), Llama-2-13B, Vicuna-7B-v1.5 (Chiang et al., 2023), Vicuna-13B-v1.5, and StableLM-7B (Taori et al., 2023). These models allow us to comprehensively evaluate the effectiveness of our factual detection method across various LLM architectures and configurations.

Experimental Environment. We conduct experiments on a server with 32 Intel Xeon Silver 4314 CPUs at 2.40 GHz, 386 GB of RAM, and an NVIDIA A100 Tensor Core GPU. Training LLM Factoscope on different LLM architectures takes under 24 hours. Results presented are from a single run, but multiple runs yield similar outcomes.

5.2 Effectiveness

In evaluating the performance of LLM Factoscope, we use activation values from specific layers to establish a comparative baseline, particularly those in the middle to later stages of the LLMs, which aligns with our observations in Figure 1. We use these activation values as input features to train a fully connected neural network for factual detection. Detailed settings are shown in Appendix A. As shown in Table 2, our LLM Factoscope con-

Table 2: Effectiveness results of LLM Factoscope across different LLM architectures.

Method	GPT2-XL-1.5B	Llama2-7B	Vicuna-7B	Stablelm-7B	Llama2-13B	Vicuna-13B
Ours	**0.961**	**0.967**	**0.982**	**0.983**	**0.983**	**0.974**
Baseline	0.880	0.888	0.831	0.817	0.882	0.785

Table 3: Generalization Performance of different LLM architectures. Abbreviations: BL - Baseline, BA - Book-Author, PC - Pantheon-Country, AC - Athlete-Country.

Data	GPT2-XL-1.5B		Llama2-7B		Vicuna-7B		Stablelm-7B		Llama2-13B		Vicuna-13B	
	Ours	BL	Ours	BL	Ours	BL	Ours	BL	Ours	BL	Ours	BL
BA	0.712	**0.800**	**0.977**	0.701	**0.971**	0.757	**0.690**	0.333	**0.904**	0.854	**0.895**	0.814
PC	0.871	**0.879**	**0.972**	0.946	**0.790**	0.516	**0.635**	0.610	**0.938**	0.849	**0.913**	0.698
AC	**0.979**	0.780	**0.703**	0.693	**0.770**	0.716	0.694	**0.756**	**0.778**	0.686	**0.807**	0.703
Average	**0.854**	0.818	**0.884**	0.780	**0.844**	0.663	**0.673**	0.566	**0.873**	0.763	**0.872**	0.738

sistently maintains high accuracy levels, ranging between 96.1% and 98.3%, across different LLM architectures. In contrast, the accuracy of the baseline fluctuates between 78.5% and 88.8%. This variation suggests that as LLMs increase in parameter size, the regions responsible for different types of factual knowledge might differ, or multiple layers could be involved in representing a single type of factual knowledge. Consequently, the baseline, which relies solely on activation values from a single layer, demonstrated unstable performance. We also provide an interpretation analysis on the LLM Factoscope in Appendix B.

Based on the analysis, we believe that the superior performance of the LLM Factoscope can be attributed to its consideration of various inner state changes across layers. By integrating this multi-dimensional analysis of inner states within LLMs, LLM Factoscope effectively discerns factual from non-factual outputs, offering a more robust and reliable approach to factual detection. This method's success not only highlights the significance of activation values in understanding LLM outputs but also paves the way for future explorations into the intricate workings of LLMs, particularly in the realm of natural language processing applications.

5.3 Generalization

It is well-established in neural network research that the effectiveness of a model largely depends on the similarity between training and testing distributions (Yang et al., 2021). Thus, our model's performance may vary across different distributions. We adopt a leave-one-out approach for our generalization assessment. Specifically, we remove one relation dataset, train the model on the remaining datasets, and then test it on the omitted dataset. We selected three relations for assessing generalization, including Book-Author, Pantheon-Country, and Athlete-Country, as these relations form sizable datasets across all LLMs tested.

Our empirical findings suggest that different LLMs exhibit varying generalization capabilities across different relations, as shown in Table 3. In the "Book-Author" relation, our method achieved a notable 97.7% accuracy with Llama2-7B and 69.0% with Stablelm-7B. This variation is likely due to each LLM's unique handling of different types of factual knowledge. Our method outperforms the baseline in most cases, with its average performance consistently outperforming the baseline, indicating its superior generalization ability. Stablelm-7B exhibits the weakest generalization among all the LLMs tested. This aligns with its relatively lower scores on LLM leaderboards (Beeching et al., 2023). We hypothesize that this could be attributed to its less effective learning of factual versus non-factual content it was trained on.

The activation maps are particularly sensitive to the type of data being processed; for instance, neurons activated in response to art-related prompts may differ from those responding to science-related prompts. In contrast, final output ranks, top-k output indices, and top-k output probabilities are agnostic to the data type. Leveraging these features to observe the LLM's decision-making evolution significantly enhances the generalization capabilities of the LLM Factoscope across diverse domains. While our LLM Factoscope shows generalization, we advise matching testing and training data distributions for optimal use. For example, an LLM used as a historical assistant should train the LLM Factoscope with historical data for best performance.

5.4 Ablation Study

Contribution of each sub-model. We evaluate the contribution of each sub-model by incrementally adding them to the factual detection model on

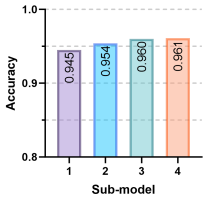

Figure 6: Effectiveness of Sub-models

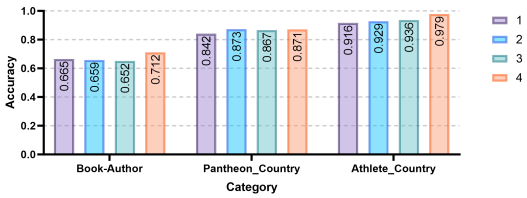

Figure 7: Generalization of Sub-models

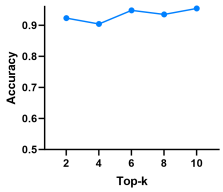

Figure 8: Effects of Top-k

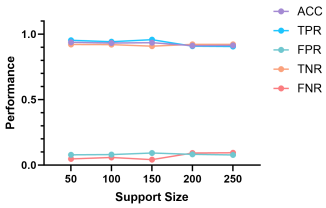

Figure 9: Effects of support set size

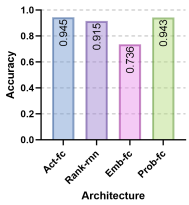

Figure 10: Effects of architecture

GPT2-XL-1.5B. As depicted in Figure 6, we notice a slight but consistent improvement in accuracy with adding more sub-models. This indicates that each sub-model brings a unique dimension to the model's capabilities, enhancing its overall performance. We employ the "leave one out" training approach as in Section 5.3 to assess the contribution of sub-models to generalization. The results in Figure 7 demonstrate enhanced generalization as more sub-models are integrated. The figures illustrate a sequential integration, starting with "1" that uses activation maps, and culminating in "4" that combines activation maps, final output ranks, top-k output indices, and top-k output probabilities. This improvement is particularly evident in the final model, which shows an increase in accuracy across various datasets compared to the model with only one sub-model.

Effects of different top-k. The top-k affects the top-k output indices and top-k output probabilities. The previous experiments set the top-k to top-10 unless otherwise stated. Now, we will evaluate the effect of choosing different values for top-k on the performance of the factual detection model. We set the k to 2, 4, 6, 8, 10 on GPT2-XL-1.5B. The results of the experiment are shown in Figure 8. The lowest performance is 90.4% when k is 4, and the highest performance is 95.4% when k is 10. The difference between the two is 5%.

Effects of different support set size. We also try different support set sizes from 50 to 250, observing their impact on the performance of the factual detection model. This evaluation was conducted on the Llama2-7B. The results, as presented in Figure 9, demonstrate that the change in support set size does not significantly impact the model's performance across most metrics.

Effects of different sub-models' architectures. We use different sub-model architectures and assess the performance of the factual detection model. We use fully connected layers to replace the ResNet18 and RNN to replace the GRU network. As shown in Figure 10, when we replace parts of the architecture with fully connected layers (act-fc, prob-fc) and RNNs (rank-rnn), we notice a slight decline in performance. In contrast, the emb-fc architecture, where we replace the ResNet18 with fully connected layers, results in a significant performance drop with accuracy falling to 73.6%. Such a drastic drop highlights the pivotal role of ResNet18 in effectively capturing the LLM's top-k output indices. The results show that while the model demonstrates resilience to certain architectural changes, some alterations can substantially impact its performance.

6 Conclusion

We discover effective inner states and provide observation for LLM factual detection. Then, we develop a factual detection pipeline, the LLM Factoscope, which leverages these inner states to detect the factuality of LLM outputs. The LLM Factoscope consistently demonstrated high factual detection accuracy, surpassing 96% in our custom-collected datasets. Our research not only provides a novel method for LLMs' factual detection but also opens new avenues for future explorations into the LLMs' inner states. By paving the way for enhanced model understanding and reliability, the LLM Factoscope sets a foundation for more transparent, accountable, and trustworthy use of LLMs in critical applications.

Limitations. Our research assumes LLMs' training corpora are predominantly factual, despite the potential inclusion of some non-factual content. Our pipeline remains effective as long as factual outweighs non-factual data. When creating our dataset, we generate various prompts and responses to simulate factual inquiries. Despite our efforts, the diversity and complexity of real-world knowledge may not be fully represented in our custom dataset. However, owners of LLMs providing factual detection services likely have access to more comprehensive and meticulously vetted datasets. Such resources could potentially enhance the performance and accuracy of LLM Factoscope. The LLM Factoscope shows generalizability in our evaluations, effectively detecting factuality across various contexts. However, it may encounter challenges when dealing with data types that significantly deviate from those in our training dataset. Following our pipeline to fine-tune the model with new, more representative datasets could further refine its accuracy and broaden its applicability.

Ethical Consideration. Our research utilizes publicly available datasets from Kaggle, which do not contain sensitive content, thereby mitigating direct concerns regarding data privacy and protection. We have adhered strictly to Kaggle's terms of use and conditions, ensuring the lawful utilization of these datasets. Although our research aims to detect the factuality of LLMs' outputs, the development and application of this technology could have broader social impacts. Positively, improving information accuracy can increase public trust in digital content and has beneficial implications for education, scientific research, and journalism by verifying the factuality of LLM outputs. However, we are also cautious of potential misuse, such as employing this technology for censorship, controlling information flow, exacerbating subjective interpretations and manipulation of "facts".

References

Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. Llm based generation of item-description for recommendation system. In Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, page 1204–1207, New York, NY, USA. Association for Computing Machinery.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. arXiv preprint arXiv:2304.13734.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

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

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases.

Nidula Elgiriyewithana. 2023. Billionaires statistics dataset. https://www.kaggle.com/datasets/nelgiriyewithana/billionaires-statistics-dataset. Accessed: 2023-10-08.

The Nobel Foundation. 2023. Nobel laureates dataset. https://www.kaggle.com/datasets/nobelfoundation/nobel-laureates. Accessed: 2023-10-08.

Daniel Grijalvas. 2023. Movies dataset. https://www.kaggle.com/datasets/danielgrijalvas/movies. Accessed: 2023-10-08.

The Guardian. 2023. Olympic games dataset. https://www.kaggle.com/datasets/the-guardian/olympic-games. Accessed: 2023-10-08.

Jinwen He, Kai Chen, Guozhu Meng, Jiangshan Zhang, and Congyi Li. 2023. Good-looking but lacking faithfulness: Understanding local explanation methods through trend-based testing. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS '23, page 431–445, New York, NY, USA. Association for Computing Machinery.

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), pages 770–778.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. Multimedia tools and applications, 82(3):3713–3744.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. Advances in Neural Information Processing Systems, 36.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Mexwell. 2023. Famous paintings dataset. https://www.kaggle.com/datasets/mexwell/famous-paintings. Accessed: 2023-10-08.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. ArXiv, abs/2302.07842.

MIT. 2023. Pantheon project dataset. https://www.kaggle.com/datasets/mit/pantheon-project. Accessed: 2023-10-08.

Edoardo Mosca, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the LLM era. In Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), pages 190–207, Toronto, Canada. Association for Computational Linguistics.

Naren. 2023. Goodreads best books dataset. https://www.kaggle.com/datasets/meetnaren/goodreads-best-books. Accessed: 2023-10-08.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra-Aimée Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. ArXiv, abs/2306.01116.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, J. Liu, Hao Tian, Huaqin Wu, Ji rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. ArXiv, abs/2307.11019.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

D. Sculley. 2023. Kaggle: Your home for data science. https://www.kaggle.com. Accessed: 2023-10-08.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In International Conference on Machine Learning.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Prashant Upadhyay. 2023. The top paying sports teams and top paid athletes dataset. https://www.kaggle.com/datasets/prashant808/the-top-paying-sports-teams-and-top-paid-athletes. Accessed: 2023-10-08.

Shriyash Upadhyay, Etan Ginsberg, and Chris Callison-Burch. 2023. Improving mathematics tutoring with a code scratchpad. In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 20–28, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Wikidata. 2023. Wikidata query service. https://query.wikidata.org/. Accessed: 2023-10-08.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. arXiv preprint arXiv:2110.11334.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, Xiang Wan, Benyou Wang, and Haizhou Li. 2023a. Huatuogpt, towards taming language models to be a doctor. arXiv preprint arXiv:2305.15075.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models.

A Training Details

We use a Siamese-based model architecture in the LLM Factoscope. The LLM Factoscope model

comprises several sub-models, each tailored to handle a specific type of inner state. This includes a ResNet18 model for processing activation values, a GRU network for final output rankings, and two additional ResNet18 models for handling top-k output indices and top-k output probabilities.

We set top-k to top-10. The output of each sub-model is an embedding of dimension 24. These embeddings from each sub-model are concatenated, resulting in a combined embedding of dimension 96. (24 dimensions from each of the four sub-models). This combined embedding is then fed into a fully connected layer, which reduces the dimensionality to 64, ensuring a compact yet informative representation. The final embedding undergoes ReLU activation and L2 normalization, providing a normalized feature vector for each input. During testing, the size of the support set is set to 100. During training, the triplet margin loss function with a margin α of 1.0. This configuration enhances the model's distinction between factual and non-factual content.

We use activation values from specific layers to establish a comparative baseline. For GPT2-XL-1.5B, the model is based on the activation values from the 31st layer. In the case of Llama2-7B and Vicuna-7B, the 23rd layer's activation values are used. For Stablelm-7B, the baseline model relied on the 12th layer, while Llama2-13B and Vicuna-13B utilize the activation values from their 32nd layers. To ensure reproducibility and transparency in our research, we have documented the key parameters relevant to our experiments in Table 4. This detailed listing of hyperparameters aims to provide comprehensive insights into the configuration settings used in our model, thereby facilitating the replication of our study and validation of our findings.

Parameter	Value
k	10
Sub-model Output Embedding Dimension	24
Combined Embedding Dimension	96
Fully Connected Layer Output Dimension	64
Activation Function	ReLU
Normalization	L2 normalization
Support Set Size	100
Epochs	30
Batch Size	64
GRU Model Hidden Dimensions	1st - 128, 2nd - 64
Dataset Split	80% training, 20% testing
Activation Dimension of GPT2-XL	6400
Activation Dimension of Llama-2-7B	11008
Activation Dimension of Vicuna-7B-v1.5	11008
Activation Dimension of Llama-2-13b	13824
Activation Dimension of StableLM-7B	24576
Optimizer	Adam
Learning Rate	0.001
Betas	0.9, 0.999
Epsilon	1e-08
Weight Decay	0
Loss Function	Triplet Margin Loss
Margin	1.0
p	2.0
Epsilon	1e-06

Table 4: Model Parameters for LLM Factoscope

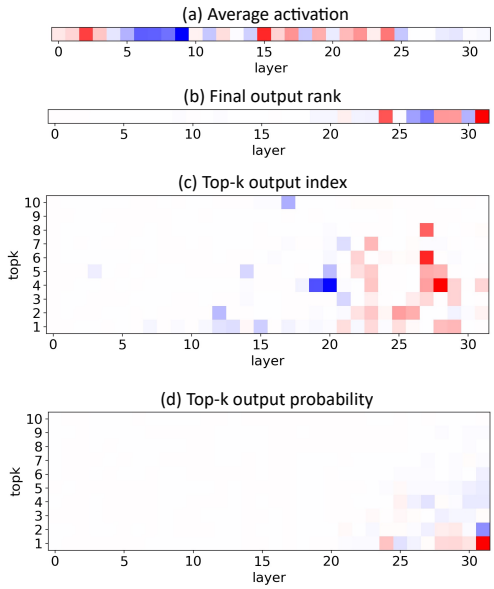

Figure 11: Visualization of different inner states' contribution.

B Interpretability

We delve into the interpretability of the LLM Factoscope, aiming to analyze the contribution of these features in discerning the factualness of LLM outputs. Specifically, we use the Integrated Gradient (Sundararajan et al., 2017) to analyze the contribution of activation maps, final output ranks, top-k output indices, and top-k output probabilities. Integrated Gradient is particularly chosen for its higher faithfulness in interpretability assessments (He et al., 2023). Our analysis reveals that the most influential features are mainly in the middle to later layers of the LLMs, consistently observed across all four data types. To provide a clearer visualization of this pattern, we present a typical example in Figure 11. In the figure, red indicates a positive contribution, while blue signifies a negative contribution, with deeper colors representing higher importance. Due to the high dimensionality of activation maps, we display the average importance of features at each layer. The majority of positive contributions emerge after the 15th layer. This finding aligns with our observations that the model initially filters semantically coherent candidate outputs in the earlier layers, and then progressively focuses on candidates relevant to the given prompt task in deeper layers.