ReFL: Reflective Feedback Learning for Hallucination Detection of Large Language Models

Cunhang Fan[1], Jun Zhang[1], Xue Zhang[1], Shuai Zhang[2], Zhao Lv[1], Jianhua Tao[2,3,*], Zhengqi Wen[3,*]

[1]State Key Laboratory of Opto-Electronic Information Acquisition and Protection Technology, (School of Computer Science and Technology), Anhui University

[2]Department of Automation, Tsinghua University

[3]Beijing National Research Center for Information Science and Technology, Tsinghua University

Abstract

Large Language Models (LLMs) often generate factually incorrect content, known as "hallucinations", which undermine the reliability and safety of their outputs. Existing hallucination detection methods either depend on external knowledge sources, incurring high computational costs and limiting real-time applicability, or extract the model's internal states, leading to poor generalization. To address these issues, this paper proposes ReFL, a hallucination detection framework. ReFL leverages corrective in-context learning to dynamically guide LLMs to recognize their own prediction errors and adjust internal representations, critically without updating model weights. Specifically, by introducing a corrective in-context learning strategy, where triplets of input text, model prediction, and ground-truth label are embedded into the prompt to make the model explicitly aware of its own errors. The model reflects on prior outputs to adjust its internal states and generate semantically structured representations better aligned with factuality. This feedback mechanism encourages the model to shape a more coherent semantic space and enhances the LLM's internal sensitivity to hallucinations. Experimental results on two benchmark datasets demonstrate that ReFL consistently outperforms existing methods, achieving state-of-the-art performance.

1 Introduction

In recent years, Large Language Models (LLMs) have developed rapidly and shown strong generative capabilities in various natural language processing tasks (Brown et al., 2020; Touvron et al., 2023a). However, LLMs often generate factually incorrect content, a phenomenon known as "hallucination" (Ji et al., 2023). These hallucinations reduce the credibility and trustworthiness of model outputs, especially when the content appears fluent and convincing. In high-stakes domains such as healthcare, law, and scientific research, hallucinations may even lead to misleading or dangerous decisions. essential across various domains (Bai et al., 2025; Fan et al., 2025a,b). Therefore, effectively detecting and identifying hallucinations has become a key challenge for improving the safety and reliability of LLMs (Guan et al., 2024).

According to Metcalfe (2017), humans learn not only from success but also from their own mistakes. When humans face a question they previously answered incorrectly, people often reflect on their reasoning process, analyze the source of the error, and make more robust judgments in similar situations. This ability of "reflection and feedback" enables humans to continuously correct cognitive biases and improve accuracy in complex tasks. Inspired by this reflective feedback mechanism, we raise a central question:

> Can LLMs also learn to detect hallucinations by "observing their own errors", and then adjust their internal reasoning patterns, thereby improving the reliability of content generation?

Our ReFL framework provides an affirmative answer by introducing a reflective feedback mechanism that enables LLMs to perform self-correction purely through dynamic prompt guidance, critically without requiring updates to the underlying model's parameters. This approach stands in contrast to existing internal state-based methods (Huang et al., 2025; Ji et al., 2024). For example, SAPLMA analyzes hidden layer activation values to assess output reliability (Azaria and Mitchell, 2023). MixHD extracts internal features such as hidden layer states and word probabilities, and analyzes the output probability distribution (Li et al., 2025). However, they generally face a key issue: most methods treat internal states as static features, lacking dynamic intervention and guidance during the reasoning

* Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19648–19665

process (Zhang et al., 2025a). They lack a self-feedback mechanism for LLMs and fail to use the model's error awareness to adjust and optimize internal representations. This passive approach limits the model's ability to capture structural information related to hallucinations.

To address these issues, this paper proposes a new hallucination detection framework called ReFL. Unlike prior methods that treat hidden states as fixed outputs, ReFL embeds correction signals into the model's prompt to guide semantic realignment. Specifically, ReFL constructs triplets consisting of "input text," "model-predicted label," and "ground-truth label" as context examples. These triplets highlight prediction errors, triggering adjustments within the internal representations, thereby allowing the model to self-adjust its reasoning process and enhance its sensitivity to factual content during inference. This reflection process is similar to how humans refine knowledge through feedback, making the model more sensitive to hallucinated outputs. This process helps the model generate more structured and accurate semantic representations. Unlike methods that require querying external knowledge bases at inference time, ReFL operates solely on the model's internal states after being trained on supervised datasets that provide ground-truth labels. By incorporating feedback from previous predictions, ReFL enables the model to focus more on factual content and reduce attention to hallucinated information, improving its hallucination detection ability across various contexts. The key contributions of the approach can be summarized as follows:

- The ReFL framework is proposed, introducing reflective feedback learning via corrective in-context learning. This mechanism dynamically guides the internal states of large language models without requiring parameter updates, significantly enhancing hallucination detection.

- A parameter-efficient feedback learning mechanism within the ReFL framework enables dynamic, real-time adjustments to the model's internal state.

- Achieving state-of-the-art performance on two benchmark datasets, the approach demonstrates the strong generalization of feedback-guided internal state modeling for hallucination detection.

2 RELATED WORKS

2.1 Hallucination in LLMs

"Hallucination" refers to factual or semantic errors that occur when LLMs generate content (Zhang et al., 2025b). This issue is prevalent in tasks such as text generation, question answering, summarization (Pagnoni et al., 2021), and dialogue (Dziri et al., 2022). Hallucinations severely affect the accuracy and trustworthiness of model outputs (Farquhar et al., 2024). As LLMs are increasingly deployed in critical applications, ensuring their factual consistency is paramount (Xu et al., 2025).

To address this challenge, researchers have proposed various hallucination detection methods (Chen et al., 2025b; Liang et al., 2025). MIND proposed an unsupervised hallucination detection method. They showed that the contextual embedding of the last token in the final layer plays a key role in hallucination detection (Su et al., 2024). PRISM guide the structural changes in the internal states of LLMs using carefully designed prompts (Zhang et al., 2025a). Some methods also detect hallucinations by analyzing token probabilities (Quevedo et al., 2025; Duan et al., 2024). Beyond analyzing the internal state of a single output, some studies explore hallucination detection through consistency among multiple generations (Agrawal et al., 2024; Mündler et al., 2024; Chen et al., 2025a). SelfCheckGPT found that the consistency among multiple responses to the same question can indicate the confidence of the LLM's output (Manakul et al., 2023). Building on this, INSIDE extended consistency analysis from textual outputs to the LLM's internal representation space. They compared the similarity between hidden vectors of multiple generations to effectively identify hallucinations (Chen et al., 2024).

2.2 Self-Correction in LLMs

In-context learning (ICL) has become a key paradigm for using large language models LLMs (Brown et al., 2020). Researchers have developed various prompting techniques, such as Chain-of-Thought reasoning (Wei et al., 2022b), Self-Reflection (Renze and Guven, 2024), and Instruction Tuning (Wei et al., 2022a), to improve the model's capabilities. Building on these efforts, researchers have begun to explore the self-correction ability of LLMs (Sanz-Guerrero and Von Der Wense, 2025; Wu et al., 2024), i.e., the capacity of models to revise or improve their outputs

based on prior errors or feedback. Current methods mainly adopt iterative generation (Madaan et al., 2023), fine-tuning with self-generated data, or optimization using reinforcement learning with external rewards (Kumar et al., 2024). These approaches focus on improving final outputs, typically through multi-round sampling or gradient-based learning, but rarely intervene in the internal representation space. They often treat model predictions as passive outputs, overlooking the potentially useful information they may contain—especially when the predictions are incorrect (Hamdan and Yuret, 2025). Such predictions, though wrong, often reflect meaningful biases that can inform internal state restructuring. We argue that even incorrect predictions can positively influence the structure of the LLM's internal states.

3 Methodology

In this section, we provide a detailed introduction to our proposed ReFL framework, which stands for reflective feedback learning for hallucination detection of LLMs. The overall process is illustrated in Figure 1. This framework consists of two steps: Corrective In-Context Learning (CICL) construction and hallucination classifier training.

3.1 Corrective In-Context Learning Generation

We adopt a standard k-shot in-context learning strategy to construct few-shot prompts for each evaluation sample x. Specifically, let $\mathcal{L}$ be the set of possible discrete labels in the classification task. We select a set of k labeled examples $C_k = \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$ to construct the context prompt, where x_i is the input statement and $y_i \in \mathcal{L}$ is the corresponding label. Given a new test instance x, the model M is expected to infer the most probable label conditioned on the prompt:

$$\hat{y} = \arg\max_{y \in \mathcal{L}} P_M(y \mid \mathcal{C}_k, x) \tag{1}$$

where $P_M(y \mid \mathcal{C}_k, x)$ denotes the probability that model M assigns to label y, given the prompt context C_k and the input sample x. To compute this probability, we follow the common practice of token-level likelihood based on the LLM's output logits, where geometric mean normalization is applied to mitigate label length bias (details can be found in Appendix E). The label with the highest normalized likelihood is then selected as the final prediction.

After obtaining the predicted label for each input text, we construct a set of corrective triplets that make explicit the agreement or discrepancy between the model's prediction and the ground truth label. To make the model "aware" of its own past errors, we define a corrective triplet $t_i = (x_i, \hat{y}_i, y_i)$, where x_i is the input text, $\hat{y}_i$ is the predicted label, and y_i is the true label. These triplets explicitly encoding the relationship between what the model believed versus what is factually correct. For incorrectly predicted samples ($\hat{y}_i \neq y_i$), the triplet functions as a negative feedback signal, helping the model identify its own hallucination-prone reasoning behaviors. For correctly predicted samples ($\hat{y}_i = y_i$), the triplet acts as a positive alignment signal, reinforcing structural indicators of factual reasoning. This subtle yet powerful feedback mechanism, embedded within the prompt context, dynamically reshapes the LLM's internal representation space at inference time, making it more sensitive to hallucinated outputs without altering its fundamental parameters. We use these triplets as semantic feedback cues to construct a new prompt for the test sample:

$$\mathcal{C}_{\text{CICL}} = \{(x_j, \hat{y}_j, y_j)\}_{j=1}^{k} \tag{2}$$

For the test sample x, we first predict its label $\hat{y}_i$ using the above ICL strategy. We then append this test input and its predicted label to the constructed context, yielding the final CICL input sequence:

$$\text{Input}_{\text{CICL}} = \mathcal{C}_{\text{CICL}} \,||\, (x, \hat{y}, ?) \tag{3}$$

where $||$ denotes sequential concatenation, and the $?$ represents the placeholder for the label to be inferred.

The sequence is fed into the same model M, allowing it to "see" the outcome of its past predictions. This process encourages the model to refine its internal state based on prior errors and simulates a contextual self-correction mechanism, where the model introspects on its prior outputs, facilitating internal realignment toward factual consistency. Our approach enables the model to process its own uncertainty (Xiong et al., 2024) and mistakes. It uses the model's own past predictions—both correct and incorrect—as semantic guidance signals to reshape internal representations.

3.2 Hallucination Classifier Training

The internal states of LLMs contain rich and dense semantic information. For the t-th output token

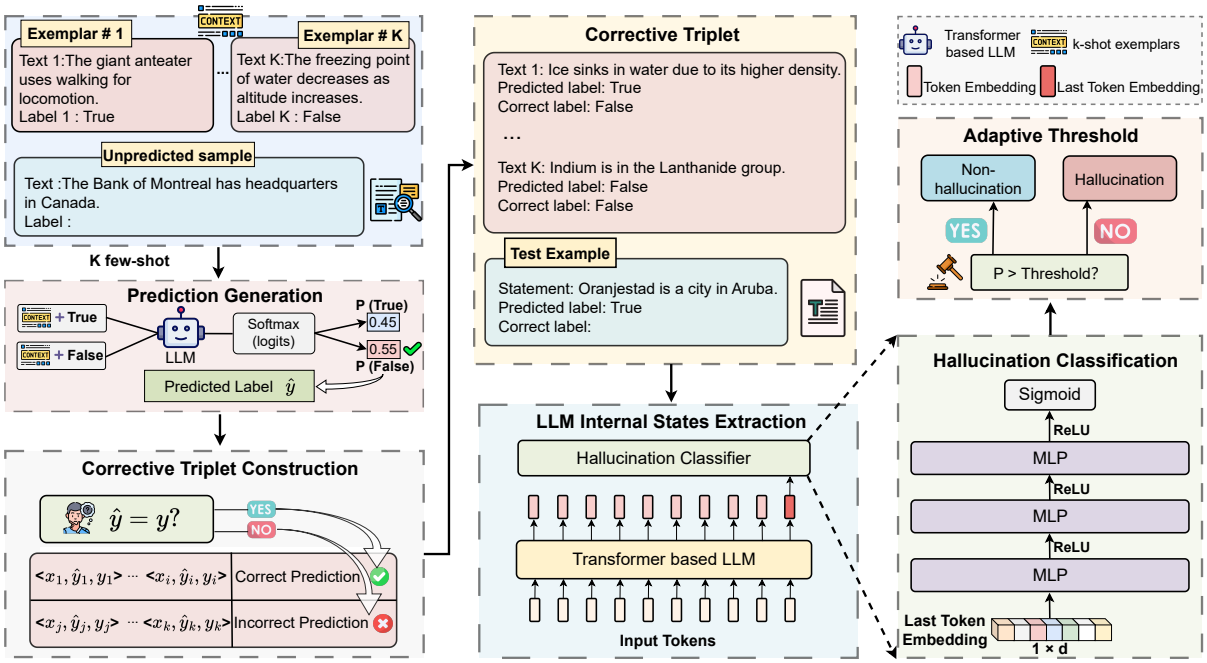

Figure 1: Overview of the ReFL Framework for Hallucination Detection in LLMs: The framework includes key components such as prediction generation, corrective triplet construction, internal state extraction, and hallucination detection. It starts with generating prediction labels via a few-shot setup, followed by comparing predictions with ground-truth labels to construct corrective triplets. These triplets, along with the test input and predicted label, are fed into the model to extract internal states. A hallucination classifier then uses these states to determine if the prediction is a hallucination.

y_t, we denote the hidden embedding at layer l as $\mathbf{h}_t^l \in \mathbf{R}^d$, where d is the dimensionality of the hidden representation at that layer. We extract the embedding of the last token from the final layer as the feature vector (Su et al., 2024). This vector can be viewed as the LLM's final understanding of the input semantics, incorporating both its own prediction and the feedback from the true label. We focus on the contextual semantic embeddings formed when the model processes each CICL triplet input, using the predicted label to enhance the LLM's ability to assess the factuality of the text.

We trains a binary classification model to determine whether a given input contains hallucinated content based on its internal representation. We use a multilayer perceptron (MLP) as the classifier. Its input is a $1 \times d$ vector, which corresponds to the hidden state of the last token from the final layer of the LLM. The output is the predicted probability $P \in [0, 1]$ for the binary classification. The forward propagation of the classifier can be represented as:

$$P = \text{MLP}(\text{ReLU}(W \cdot H + b)) \quad (4)$$

where W and b represent the weight matrices and bias terms, respectively. The MLP consists of three hidden layers with dimensions 256, 128, and 64. Each hidden layer uses the ReLU activation function. The output layer uses a sigmoid function to map the result to a probability between 0 and 1. To train the classifier, we use the Binary Cross-Entropy (BCE) loss function as the optimization objective. It is defined as follows:

$$\mathcal{L}_{\text{BCE}}(y, p) = -y \log(p) - (1-y) \log(1-p) \quad (5)$$

where $y \in [0, 1]$ denotes the ground truth label of the sample, where 1 indicates factual content (non-hallucinated), and 0 indicates the presence of hallucination. p is the classifier's predicted probability that the sample is factual. By minimizing this loss function, the classifier learns to map embedding-level differences into hallucination predictions, enabling the model to distinguish internal representations of truthful vs. hallucinated samples.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets.

We evaluate the performance of ReFL and baseline models on the True-False dataset (Azaria and Mitchell, 2023) and the LogicStruct dataset (Bürger

et al., 2024; Marks and Tegmark, 2024). The True-False dataset is designed to study cross-domain hallucination detection. Each example is a short factual statement labeled as true or false, covering six semantically independent topics: animals, cities, companies, chemical elements, scientific facts, and inventions.

To further investigate hallucination detection under variations in syntactic structure, we use the LogicStruct dataset. It consists of 24 subsets across six topics. Each topic contains four syntactic variants: affirmative, negation, conjunction, and disjunction. Negation sentences are constructed by inserting the word "not" into affirmative statements. Conjunctive and disjunctive sentences are formed by sampling two affirmative clauses and connecting them with "and" or "or" to create more complex composite statements. More dataset information and details can be found in Appendix A.

4.1.2 Models and Evaluation Metrics.

In our experiments, we utilize a diverse set of representative open-source LLMs, encompassing various architectures and scales. For the main results presented in the paper, we focus on LLaMA2-7B (Touvron et al., 2023b), LLaMA3-8B (Grattafiori et al., 2024), Qwen2.5-3B-instruct (Qwen et al., 2025), and OPT-6.7b (Zhang et al., 2022). To further demonstrate ReFL's robustness and generalizability, additional experiments were conducted on other models, with their results detailed in Appendix B. All models are obtained via Hugging Face and used in their base or instruction-tuned versions (as specified), without additional fine-tuning.

To assess the effectiveness of different hallucination detection methods, we adopt two standard evaluation metrics: Accuracy (ACC) and Area Under the Receiver Operating Characteristic Curve (AUC).

4.1.3 Baseline.

We select the following hallucination detection methods as baselines, with all experimental results reproduced by ourselves for consistent evaluation:

- **Predictive Probability (PP)** (Manakul et al., 2023) is based on the token-level probabilities assigned by the LLM during generation. It measures the model's confidence in its own output. The variant Length-Normalized Predictive Probability (LN-PP), which averages the log-probabilities of all tokens.
- **SAPLMA** (Azaria and Mitchell, 2023) extracts activation values from LLM hidden layers as feature vectors and trains a classifier to assess the factuality of text statements. SAPLMA achieves stable classification performance across domains.
- **MIND** (Su et al., 2024) uses pseudo-labeled data extracted from high-quality Wikipedia corpora to train a classifier based on LLM internal states, enabling real-time detection without manual annotation.
- **MM** (Marks and Tegmark, 2024) quantifies factuality by projecting a test sample's embedding onto a "factuality direction." A sigmoid function maps this projection to a probability for hallucination detection.
- **PRISM** (Zhang et al., 2025a) uses carefully designed prompts to guide the internal states of LLMs. It integrates with existing methods such as SAPLMA and MM, resulting in PRISM-SAPLMA and PRISM-MM.

4.2 Implementation Details

On the True-False dataset, we set the number of few-shot examples to k = 6 for LLaMA2-13B and LLaMA3-8B, and k = 4 for all other models. On the LogicStruct dataset, we used k = 4 for all models. To ensure balanced feedback signals in the prompt, we fix the correction ratio to 50%. Specifically, prediction samples are divided into a Correct Prediction pool ($\hat{y} = y$) and an Incorrect Prediction pool ($\hat{y} \neq y$), from which $k/2$ exemplars are randomly sampled from each pool to construct the prompt for every test instance.

For each topic in the True-False dataset, we train a classifier using activation values from all other topics and test it on the current topic to evaluate accuracy under a cross-topic setting. For the LogicStruct dataset, we train the model on affirmative sentences and test it on negative, conjunction, and disjunction sentences to assess generalization across different syntactic structures (Levinstein and Herrmann, 2024). The MIND method is trained on its original automatically constructed dataset and tested on True-False and LogicStruct datasets. For models that require a validation set, we split the training data into training and validation subsets at a 4:1 ratio and select the best-performing parameters on the validation set for final evaluation. Complete hyper parameter configurations and implementation details are provided in Appendix B to ensure full reproducibility.

Qwen-3B-Instruct ACC (%)								OPT-6.7B ACC (%)						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**
LN-PP	57.2	55.2	52.8	59.0	55.0	59.7	56.5	52.5	53.9	52.4	54.0	53.5	55.9	53.7
MIND	48.5	47.7	50.7	50.1	50.1	51.5	49.8	49.9	48.2	47.2	50.1	49.9	41.6	47.8
MM	53.4	75.3	63.3	54.2	66.7	68.6	63.6	50.3	51.0	49.3	51.1	57.6	47.5	51.1
SAPLMA	73.9	89.7	67.3	62.3	84.8	78.6	76.1	64.9	51.2	63.7	56.3	67.1	57.1	60.0
PRISM[1]	79.3	91.2	82.9	79.3	**92.5**	84.4	84.9	50.0	50.3	**67.4**	50.2	49.9	53.9	53.6
PRISM[2]	81.1	91.2	87.6	82.3	85.8	**88.8**	86.1	**68.4**	62.8	61.9	61.5	63.9	57.9	62.7
ReFL	**84.6**	**92.2**	**89.0**	**83.1**	92.3	87.9	**88.2**	59.2	**74.2**	62.3	**63.1**	**72.3**	**64.4**	**65.9**
LLaMA2-7B ACC (%)								**LLaMA3-8B ACC (%)**						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**
LN-PP	57.8	56.8	53.4	62.9	65.3	58.1	59.1	56.1	52.9	55.2	57.9	56.4	57.3	55.9
MIND	49.6	50.1	46.2	48.4	46.3	46.8	47.9	49.9	50.0	50.6	50.0	50.1	44.4	49.2
MM	50.0	53.4	52.9	50.5	67.9	49.5	54.0	60.2	52.3	52.7	56.8	62.5	59.7	57.4
SAPLMA	**73.1**	71.5	66.1	70.7	80.4	81.9	74.0	70.3	70.8	64.8	80.6	78.0	74.1	73.1
PRISM[1]	52.1	59.3	61.8	57.0	53.7	54.1	56.3	61.3	77.0	65.6	70.8	71.6	64.8	68.5
PRISM[2]	71.4	84.6	75.8	72.7	71.6	80.4	76.1	78.8	91.2	73.0	77.2	83.7	81.2	80.8
ReFL	72.6	**93.6**	**90.6**	**78.1**	**88.3**	**87.2**	**85.0**	**83.4**	**93.1**	**89.2**	**86.1**	**90.3**	**86.8**	**88.1**

Table 1: Overview of experimental results on the True-False dataset using various representative open-source LLMs. The table compares the ACC metrics for each semantic topic. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA. All numbers are percentages.

Qwen-3B-Instruc AUC (%)								OPT-6.7B AUC (%)						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**
LN-PP	60.6	68.2	58.7	67.2	68.9	64.2	64.6	52.7	66.3	59.2	56.6	65.2	58.6	59.8
MIND	48.5	61.8	48.5	62.7	58.4	49.8	55.0	41.8	46.4	52.6	51.6	38.8	39.2	45.1
MM	53.8	75.8	63.5	54.3	67.0	68.4	63.8	59.0	67.4	56.4	62.9	74.0	63.7	63.9
SAPLMA	81.3	96.2	81.2	86.6	93.0	87.8	87.7	71.3	79.6	74.1	64.8	79.2	61.5	71.8
PRISM[1]	80.6	91.6	83.3	80.0	93.0	86.3	85.8	56.5	54.4	75.1	55.0	51.6	60.9	58.9
PRISM[2]	91.1	97.6	93.8	90.8	97.8	**95.7**	94.5	**74.4**	80.8	**82.9**	67.1	76.5	68.8	**75.1**
ReFL	**93.1**	**97.9**	**94.3**	**91.9**	**99.0**	94.6	**95.1**	65.7	**83.1**	79.4	**67.5**	**82.0**	**70.3**	74.6
LLaMA2-7B AUC (%)								**LLaMA3-8B AUC (%)**						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**
LN-PP	60.4	63.8	59.4	68.1	74.8	62.6	64.9	59.7	66.2	62.7	66.5	71.5	63.5	65.0
MIND	47.6	45.6	46.0	45.9	40.3	45.9	45.2	47.2	48.4	48.6	46.0	43.3	48.2	46.9
MM	51.7	58.5	55.4	50.7	71.1	55.2	57.1	66.7	53.9	52.9	57.4	63.7	62.5	59.5
SAPLMA	82.2	89.5	78.4	78.7	88.9	91.3	84.8	81.6	77.5	78.7	88.6	90.7	85.9	83.8
PRISM[1]	52.4	61.7	68.1	56.7	54.5	52.9	57.7	62.4	80.9	69.9	75.0	76.6	64.9	71.6
PRISM[2]	79.9	93.9	83.6	76.9	88.9	87.9	85.2	87.3	97.1	92.1	89.8	93.5	88.4	91.4
ReFL	**86.4**	**98.0**	**95.7**	**89.3**	**95.6**	**94.9**	**93.3**	**91.7**	**97.5**	**96.0**	**93.4**	**98.8**	**96.1**	**95.6**

Table 2: Overview of experimental results on the True-False dataset using various representative open-source LLMs. The table compares the AUC metrics for each semantic topic. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA. All numbers are percentages.

4.3 Experimental Results

4.3.1 Hallucination Detection on the True-False Dataset

In this section, we systematically evaluate the performance of the proposed ReFL framework on hallucination detection. As shown in Tables 1 and Table 2, our method consistently achieves the best results across diverse topics and model architectures. Compared to existing baselines, ReFL demonstrates a significant performance margin across all evaluated model sizes. This substantial improvement highlights the broad effectiveness of integrating corrective in-context learning, proving that our dynamic approach consistently outperforms traditional static internal state probing methods across various evaluation settings.

Notably, on the more advanced LLaMA3-8B

Baselines	Qwen-3B-Instruct ACC (%)				LLaMA2-7B ACC (%)				LLaMA3-8B ACC (%)			
	Neg.	Conj.	Disj.	Average	Neg.	Conj.	Disj.	Average	Neg.	Conj.	Disj.	Average
LN-PP	42.0	52.2	53.3	49.2	41.8	55.5	59.4	52.2	43.2	53.0	59.9	52.0
MIND	48.0	48.9	49.7	48.9	51.6	48.1	48.9	49.5	50.6	48.0	50.9	49.8
MM	57.0	51.7	49.8	52.8	46.5	58.4	49.7	51.6	48.3	58.9	52.4	53.2
SAPLMA	77.9	75.1	52.4	68.5	45.3	50.9	50.2	48.8	48.9	59.3	50.8	53.0
PRISM[1]	84.0	88.4	54.4	75.6	50.3	50.8	50.0	50.4	47.3	60.7	51.4	53.1
PRISM[2]	85.3	87.6	57.3	76.8	53.3	**68.0**	63.0	61.4	50.0	83.8	50.7	61.5
ReFL	**92.1**	**92.3**	**68.9**	**84.5**	**72.3**	66.5	**66.1**	**68.3**	**90.9**	**84.2**	**68.1**	**81.1**

Table 3: This table shows the accuracy performance on the LogicStruct dataset using various representative open-source LLMs. Models were trained on affirmative structures and evaluated on negation, conjunction, and disjunction, with the average presented. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA.

model, our method achieves the highest average ACC of 88.1% and an AUC of 95.6%. This indicates that more powerful pre-trained language models can produce richer internal features, which are highly suitable for hallucination detection when appropriately guided. Overall, the experimental results strongly validate our core hypothesis: by explicitly injecting the discrepancy between the model's own output and the ground-truth label, LLMs can learn internal representations with strong "error-correction capability." Rather than passively observing static features, the ReFL framework forces the model to proactively reshape its internal state, allowing it to effectively and sharply distinguish factual from hallucinated content.

4.3.2 Hallucination Detection on the LogicStruct Dataset

According to Levinstein and Herrmann (2024), existing baseline models struggle to generalize from training on affirmative sentences to other syntactic structures. The fundamental reason for this poor generalization is that existing methods extract the model's internal states through standard forward propagation, and in these states, factuality features are often deeply entangled with topic and semantic features. For domain-transfer failures, existing probe models often overfit to domain specific content rather than factuality itself. When training a probe in a specific domain, the classifier often learns spurious correlations, such as binding certain keywords to the positive label, rather than the essential concept of factuality. For structural-transfer failures, taking the sentence "Paris is not the capital of France" as an example, a standard probe focuses only on the strong association between "Paris" and "France," thus predicting true, while ignoring the syntactic operator "not." Therefore, when the data distribution shifts to a new domain or new syntactic structure, these semantic dependent "shortcuts" will no longer be effective, ultimately leading to a significant drop in generalization performance.

To further evaluate the generalization ability of our method across unseen syntactic structures, we conducted a systematic evaluation on the LogicStruct dataset. As shown in Table 3, our method significantly outperforms existing approaches across all syntactic variants. Notably, on the most challenging "negation" structure, our method achieved over 90% accuracy with both the LLaMA3-8B and Qwen2.5-3B-Instruct models, substantially outperforming all baselines. Moreover, we observed that all other methods performed poorly on negation sentences for the base models without fine-tuning, while our method still demonstrated strong performance. ReFL addresses this issue by injecting corrective triplets. The core is to explicitly expose the conflict (or alignment) relationship between the model's prediction and the ground-truth label. By prompting the model with such triplets, ReFL forces the attention mechanism to focus on this universal "correctness structure" rather than semantic information specific to a particular domain or grammar. Overall, although the model was trained only on the simplest "affirmative" structure, the corrective context introduced by the ReFL framework effectively guided the model to learn hallucination-related semantic distinctions. These distinctions were then successfully transferred to more complex syntactic patterns. This improved the model's sensitivity and generalization ability across different syntactic forms. To further validate that our corrective feedback mechanism restructures the internal representation space, we conduct both quantitative and visualization analyses of the internal states, as detailed in Appendix D.

4.3.3 Efficiency Analysis

We further analyze the computational efficiency of different hallucination detection methods. All experiments are conducted under the same hardware setting using a single NVIDIA H100 GPU. For all baseline methods, we adopt their standard implementations to reflect typical usage scenarios.

The efficiency advantage of ReFL primarily stems from its architectural design. Unlike baseline methods that rely on iterative prompting or sample wise sequential processing, ReFL requires only a single forward pass through the large language model to extract feature representations. This single pass property naturally enables batch inference. In our experiments, we leverage parallel embedding extraction and the SDPA operator to fully utilize hardware throughput. Table 4 reports the feature extraction time and training time for all methods. The results show that feature extraction constitutes the primary computational bottleneck across all approaches, and ReFL achieves the lowest overall runtime among the compared methods. This efficiency gain arises from ReFL's weight update free design and its strong compatibility with parallel inference.

	Time (s)		
Baseline	**Feature Extraction**	**Training**	**Total**
MIND	270.74	71.60	342.34
MM	185.83	46.68	232.51
SAPLMA	79.04	64.47	143.51
PRISM-MM	190.78	**45.43**	236.21
PRISM-SAPLMA	190.60	119.35	309.95
ReFL	**68.43**	60.48	**128.91**

Table 4: Runtime comparison on the True-False dataset using LLaMA2-7B. Feature generation time includes all LLM forward passes required to extract representations.

4.4 Ablation Studies

4.4.1 Impact of Corrective Triplets

We explicitly compare corrective triplets with two baseline approaches: standard in-context learning (ICL) using simple input label pairs (x, y), and prior internal state probing methods that treat hidden states as static features. As shown in Table 5, ReFL consistently outperforms both baselines across most domains on the True-False dataset. Compared to standard ICL without model predictions, ReFL improves the average accuracy by approximately 4.1 percentage points, indicating the limitations of supervision based solely on correct examples. More notably, ReFL achieves an improvement of over 11 percentage points relative to the static probing setting without ReFL. These results suggest that the core advantage of ReFL lies not merely in leveraging internal states, but in actively restructuring them through corrective feedback, rather than treating them as fixed representations.

Method	**Anim.**	**Cits.**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Avg.**
Static Probe	**73.1**	71.5	66.1	70.7	80.4	81.9	74.0
Standard ICL	69.7	88.7	89.2	74.8	76.4	86.9	80.9
ReFL	72.6	**93.6**	**90.6**	**78.1**	**88.3**	**87.2**	**85.0**

Table 5: Experimental results on the True-False dataset using LLaMA2-7B. We compare ReFL with Static Probe, which directly probes internal states without prompts, and Standard ICL, which uses in-context examples composed of standard input label pairs (x, y).

4.4.2 Impact of Internal State Probing

To validate the design choice of training a classifier on the model's internal states, we fix the corrective in-context learning (CICL) mechanism and compare ReFL with a direct generative baseline. Specifically, we evaluate two approaches: (1) LLM-as-judge, where the model outputs a textual factuality label after receiving corrective in-context examples, and (2) internal state probing, which corresponds to ReFL and trains a linear classifier on the hidden representation of the last Transformer layer extracted from the same corrective input. All other experimental settings are kept identical to ensure a fair comparison.

As shown in Table 6, on the True-False dataset with LLaMA2-7B, internal state probing significantly outperforms the generative judgment approach, achieving an average accuracy of 85% compared to 61% for LLM-as-judge. This result indicates that factuality related signals are more reliably and richly encoded in the model's hidden states. During language generation, part of this information is compressed or lost, making explicit internal state probing a more effective interface for hallucination detection.

4.4.3 Impact of Correction Ratio in Prompts

We further evaluated how the proportion of corrected examples in the prompt affects model performance. Specifically, we varied the correction ratio from 0% to 100% in steps of 25% to measure how the strength of feedback affects the model's

	True-False ACC (%)						
Method	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Inv.**	**Avg.**
LLM-as-judge	66.2	46.4	71.1	63.1	62.0	57.0	61.0
Internal State	**72.6**	**93.6**	**90.6**	**78.1**	**88.3**	**87.2**	**85.0**

Table 6: Experimental results on the True-False dataset using LLaMA2-7B, comparing the internal state probing against a direct generative baseline(LLM-as-Judge).

adjustment ability. As shown in Table 7, the model achieves the best performance when the correction ratio is 50%, indicating an optimal trade-off between error signal and semantic stability. When the correction ratio is 0% (i.e., no predictive feedback), the performance is the lowest, further confirming the importance of prediction feedback. As the correction ratio increases to 75% or 100%, model performance declines because too much feedback from incorrect examples introduces noise, making it harder to generalize from reliable cues.

Ratio	**0%**	**25%**	**50%**	**75%**	**100%**
Accuracy	82.8	84.2	**85.0**	83.0	83.3

Table 7: Experimental results on the True-False dataset using LLaMA2-7B, showing the impact of different correction ratios on model performance.

4.4.4 Impact of Real-world QA Scenarios

To evaluate the effectiveness of the proposed hallucination detection method in real-world question answering scenarios, we conduct experiments on the TruthfulQA dataset (Lin et al., 2022). This dataset contains fact seeking questions collected from a variety of real-world domains that are particularly prone to hallucinated responses. We use the LLaMA2-7B-Chat model with greedy decoding to generate answers, and use BLEURT score thresholds of 0.25 and 0.5 to obtain ground-truth labels. Responses with BLEURT scores above the threshold are labeled as correct, while those below the threshold are regarded as hallucinations.

For the MM, SAPLMA, and PRISM methods, the classifier is trained on the first 20% of the dataset and evaluated on the remaining 80%. The ReFL method follows the same experimental configuration described in previous sections. Due to the imbalance in label distribution, we adopt the AUROC as the evaluation metric.

In addition, we include two unsupervised hallucination detection methods for comparison. SelfCheckGPT (Manakul et al., 2023) detects hallucinations by measuring the consistency of model responses. Specifically, we adopt the N-gram-based variant of SelfCheckGPT and evaluate two aggregation strategies: Max($-\log P$) and Avg($-\log P$). The semantic entropy (Farquhar et al., 2024) method assesses output uncertainty by computing the entropy over semantically clustered generations. For both baselines, we strictly follow their original settings and generate 20 stochastic responses with a temperature of $T = 1.0$. It is worth noting that these two methods could not be applied in earlier experiments because the input format was not in a question answer form.

As shown in Table 8, the proposed framework maintains strong performance even under more complex real-world hallucination scenarios and consistently outperforms SelfCheckGPT. In contrast, the semantic entropy method performs poorly on the TruthfulQA dataset. We hypothesize that this behavior arises from the inherent characteristics of the TruthfulQA dataset, where many correct answers are naturally diverse or refusal-based. In such cases, high semantic entropy does not indicate hallucination, but instead reflects legitimate answer diversity. Additional ablation studies and analyses are reported in Appendix C.

Baseline	**T = 0.25**	**T = 0.5**
MM	59.3	56.2
SAPLMA	75.8	54.5
PRISM-MM	56.6	58.1
PRISM-SAPLMA	54.0	63.1
Semantic Entropy	38.4	46.3
SelfCheckGPT(Avg)	81.2	44.5
SelfCheckGPT(Max)	80.6	46.5
ReFL	**91.9**	**63.3**

Table 8: Experimental results on the TruthfulQA dataset using LLaMA2-7B-chat,with different thresholds.

5 Conclusion

In this paper, we introduced ReFL, a novel hallucination detection framework for LLMs powered by reflective feedback learning. By embedding corrective feedback into in-context learning prompts, ReFL enables LLMs to dynamically self-adjust their internal representations in response to prediction-label mismatches, critically without updating model parameters. This mechanism significantly enhances hallucination sensitivity and factual alignment.

Limitations

While ReFL significantly advances hallucination detection, this study still has some limitations. Our current framework primarily focuses on statement-level factuality assessment. Expanding ReFL to tackle more complex tasks like paragraph-level hallucination detection, which demand deeper contextual understanding, is a compelling direction for future research. Additionally, ReFL's core mechanism relies on effectively utilizing LLM internal states. While effective for open-source models, applying ReFL to closed-source commercial LLMs (e.g., GPT-4o, Gemini) is challenging due to limited internal state access. Future work will explore proxy models or indirect probing techniques to address this.

Ethics Statement

In developing and evaluating the ReFL framework for hallucination detection in Large Language Models (LLMs), we are guided by the principle of enhancing the trustworthiness and reliability of AI-generated content. ReFL aims to mitigate the risks of misinformation and promote accountability in AI systems by enabling the robust detection of factual errors, thereby contributing to safer and more beneficial AI applications. Our research strictly utilizes publicly available datasets and open-source LLMs. This ensures that our work does not involve any personally identifiable information or proprietary data, thus upholding user privacy and data security standards.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) (No.62571002, 62476004), Excellent Youth Foundation of Anhui Scientific Committee (No. 2408085Y034).

References

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Kalai. 2024. Do language models know when they're hallucinating references? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 912–928, St. Julian's, Malta. Association for Computational Linguistics.

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

Jiaqi Bai, Hongcheng Guo, Zhongyuan Peng, Jian Yang, Zhoujun Li, Mohan Li, and Zhihong Tian. 2025. Mitigating hallucinations in large vision-language models by adaptively constraining information flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23442–23450.

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, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877 – 1901, Red Hook, NY, USA. Curran Associates Inc.

Lennart Bürger, Fred A Hamprecht, and Boaz Nadler. 2024. Truth is universal: Robust detection of lies in llms. *Advances in Neural Information Processing Systems*, 37:138393–138431.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, pages 1–21.

Kedi Chen, Qin Chen, Jie Zhou, Xinqi Tao, Bowen Ding, Jingwen Xie, Mingchen Xie, Peilong Li, and Zheng Feng. 2025a. Enhancing uncertainty modeling with semantic graph for hallucination detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23586–23594.

Yuyan Chen, Zehao Li, Shuangjie You, Zhengyu Chen, Jingwen Chang, Yi Zhang, Weinan Dai, Qingpei Guo, and Yanghua Xiao. 2025b. Attributive reasoning for hallucination diagnosis of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23660–23668.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M Ponti, and Siva Reddy. 2022. Faithdial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Cunhang Fan, Wang Xiang, Jianhua Tao, Jiangyan Yi, and Zhao Lv. 2025a. Cross-modal knowledge distillation with multi-stage adaptive feature fusion for speech separation. *IEEE Transactions on Audio, Speech and Language Processing*, 33:935–948.

Cunhang Fan, Hongyu Zhang, Qinke Ni, Jingjing Zhang, Jianhua Tao, Jian Zhou, Jiangyan Yi, Zhao Lv, and Xiaopei Wu. 2025b. Seeing helps hearing: A multimodal dataset and a mamba-based dual branch parallel network for auditory attention decoding. *Information Fusion*, 118:102946.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xinyan Guan, Yanjiang Liu, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2024. Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18126–18134.

Shadi Hamdan and Deniz Yuret. 2025. How much do llms learn from negative examples? *Preprint*, arXiv:2503.14391.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104, Miami, Florida, US. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):1–38.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning. *Preprint*, arXiv:2409.12917.

Benjamin A Levinstein and Daniel A Herrmann. 2024. Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*, pages 1–27.

Chuang Li, Bingnan Xing, Dongdong Huo, Qihui Zhou, Zhen Xu, and Yu Wang. 2025. Mixhd: A method for detecting hallucinations based on the internal state and output probability of large language models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Tian Liang, Yuetian Du, Jing Huang, Ming Kong, Luyuan Chen, Yadong Li, Siye Chen, and Qiang Zhu. 2025. Mole: Decoding by mixture of layer experts alleviates hallucination in large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18684–18692.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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, Singapore. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, pages 1–22.

Janet Metcalfe. 2017. Learning from errors. *Annual review of psychology*, 68(1):465–489.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*, pages 1–30.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. pages 4812–4829, Online. Association for Computational Linguistics.

Ernesto Quevedo, Jorge Yero Salazar, Rachel Koerner, Pablo Rivas, and Tomas Cerny. 2025. Detecting hallucinations in large language model generation: A token probability approach. In *Artificial Intelligence and Applications*, pages 154–173, Cham. Springer Nature Switzerland.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan

Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance, 2024. *Preprint*, arXiv:2405.06682.

Mario Sanz-Guerrero and Katharina Von Der Wense. 2025. Corrective in-context learning: Evaluating self-correction in large language models. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 24–33, Albuquerque, New Mexico. Association for Computational Linguistics.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14379–14391, Bangkok, Thailand. Association for Computational Linguistics.

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. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv: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, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, pages 1–46.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024. Large language models can self-correct with key condition verification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12846–12867, Miami, Florida, USA. Association for Computational Linguistics.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, pages 1–29.

Derong Xu, Xinhang Li, Ziheng Zhang, Zhenxi Lin, Zhihong Zhu, Zhi Zheng, Xian Wu, Xiangyu Zhao, Tong Xu, and Enhong Chen. 2025. Harnessing large language models for knowledge graph question answering via adaptive multi-aspect retrieval-augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25570–25578.

Fujie Zhang, Peiqi Yu, Biao Yi, Baolei Zhang, Tong Li, and Zheli Liu. 2025a. Prompt-guided internal states for hallucination detection of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21806–21818, Vienna, Austria. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Ahn Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025b. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, pages 1–45.

A Dataset Details

Here are some examples of statements from the True-False dataset:

- Animals: "The dog has a strong sense of smell and is known for its loyalty to humans."
- Cities: "Charlotte Amalie is a city in United States Virgin Islands."
- Companies: "Costco Wholesale has headquarters in United States."
- Elements: "Copper is a strong, lightweight metal used in aerospace and medical implants."
- Facts: "The largest lake in the world is located in Antarctica."
- Inventions: "Charles Wheatstone invented the phonograph (independent inventor)."

The displayed statements include both true and false information; specifics are provided in Table 9.

Topic	#Sentence	#T=1	#T=0
Animals	1008	504	504
Cities	1458	729	729
Companies	1200	600	600
Elements	930	465	465
Facts	613	306	307
Inventions	876	464	412

Table 9: Distribution of true (T=1) and false (T=0) statements across six subsets of the dataset, with each subset representing a different topic.

The LogicStruct dataset includes sentences with varied grammatical structures, such as:

- Affirmative statements: "The Earth's atmosphere protects us from harmful radiation from the sun."
- Negated statements: "The Earth's atmosphere doesn't protect us from harmful radiation from the sun."
- Logical conjunctions: It is the case both that [statement 1] and that [statement 2].
- Logical disjunctions: It is the case either that [statement 1] or that [statement 2].

For conjunctions and disjunctions, two statements on the same topic are combined to form complex sentences. In conjunctions, the statements are linked with 'and' to establish a logical relationship, while in disjunctions, they are connected with 'or'. Both types are derived from affirmative statements and sampled to maintain label balance across the datasets. Additional information is presented in Table 10.

Grammatical Structure	#Sentence	#T=1	#T=0
Affirmative statements	3167	1606	1561
Negated statements	3167	1561	1606
Logical conjunctions	3998	2041	1957
Logical disjunctions	3000	1494	1506

Table 10: Distribution of true (T=1) and false (T=0) statements across four grammatical structure of the dataset. Each grammatical structure is composed of six topics(animal_class, cities, inventors, element_symb, facts, and sp_en_trans). #Sentence represents the total number of data across all topics for each grammatical structure.

B More Experimental

B.1 More Implementation Details

All experiments are implemented using PyTorch and TensorFlow. Unless otherwise specified, all experiments are conducted on a single NVIDIA A800 GPU. For the ReFL method, feature representations are extracted from the final hidden layer of the language model, specifically using the embedding of the last token as the input feature to the classifier. During feature extraction, all language model parameters are kept frozen, and no weight updates are performed. Left padding is used during label prediction, while right padding is applied during embedding extraction to ensure accurate alignment of the last valid token.

On the True-False dataset, the hallucination detector adopts a lightweight multilayer perceptron architecture consisting of three hidden layers with 256, 128, and 64 units, followed by a sigmoid output layer. ReLU activation functions are applied after each hidden layer. The classifier is trained using the Adam optimizer with a learning rate of 1e-3 and binary cross-entropy loss for 30 training epochs. On the LogicStruct dataset, we employ a structurally similar multilayer perceptron, also with hidden dimensions of 256, 128, and 64 and a binary classification output. To address label imbalance in this dataset, weighted cross-entropy loss is

	True-False ACC (%)							True-False AUC (%)						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**
LN-PP	59.3	55.2	54.4	65.0	64.4	61.2	59.9	62.4	60.2	60.7	69.4	74.0	64.9	65.3
MIND	50.0	47.1	48.7	49.9	49.9	43.4	48.2	47.3	39.4	45.5	38.9	41.2	46.3	43.1
MM	74.2	50.0	50.0	64.7	76.8	67.9	63.9	79.1	66.4	55.7	73.0	86.0	72.9	72.2
SAPLMA	75.1	85.1	70.9	72.4	80.9	70.1	75.8	83.9	94.2	82.1	84.0	92.9	91.0	88.0
PRISM[1]	52.8	57.9	58.8	55.8	51.1	54.8	55.2	56.2	66.1	72.1	61.1	58.7	58.9	62.2
PRISM[2]	72.8	89.4	78.9	76.1	73.1	72.0	77.1	83.1	95.2	89.6	82.6	94.4	83.2	88.0
ReFL (ours)	**79.2**	**90.9**	**86.4**	**79.7**	**87.6**	**89.4**	**85.5**	**89.2**	**97.3**	**93.4**	**86.9**	**96.6**	**95.5**	**93.1**

Table 11: Overview of experimental results on the True-False dataset using LLaMA2-13B. The table compares the ACC and AUC metrics for each semantic topic. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA.

	True-False ACC (%)							True-False AUC (%)						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**
LN-PP	56.7	52.7	54.1	55.5	54.3	58.2	55.3	60.3	64.9	59.6	67.4	69.0	64.2	64.2
MIND	49.2	26.0	43.0	48.6	48.6	47.6	43.8	42.4	15.0	40.1	37.9	35.3	46.3	36.2
MM	56.1	45.9	63.6	59.9	75.0	64.8	60.9	56.1	46.0	63.6	60.0	75.1	62.8	60.6
SAPLMA	74.6	80.5	72.9	79.2	84.5	76.9	78.1	83.9	92.1	83.0	83.5	94.5	87.3	87.4
PRISM[1]	82.5	91.8	85.2	81.6	**95.9**	82.1	86.5	82.9	91.8	85.3	82.2	96.4	83.0	86.9
PRISM[2]	83.7	91.9	90.0	78.5	92.0	86.9	87.2	**93.5**	98.3	**95.2**	90.5	99.0	**95.5**	95.4
ReFL (ours)	**84.4**	**94.1**	**90.5**	**86.3**	95.0	**87.5**	**89.6**	93.4	**98.6**	94.6	**92.8**	**99.3**	95.4	**95.7**

Table 12: Overview of experimental results on the True-False dataset using Qwen2.5-7B-instruct. The table compares the ACC and AUC metrics for each semantic topic. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA.

	True-False ACC (%)							True-False AUC (%)						
Baselines	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**	**Anim.**	**Cities**	**Comp.**	**Elem.**	**Facts**	**Invent.**	**Average**
LN-PP	56.4	52.8	56.8	57.7	55.8	59.7	56.5	59.2	63.0	60.9	67.4	69.3	65.3	64.2
MIND	49.7	50.0	46.9	50.0	49.9	44.1	48.4	43.7	49.7	41.1	37.5	35.5	42.2	41.6
MM	58.3	52.8	58.1	61.7	65.9	62.0	59.8	58.8	52.1	58.4	61.6	66.4	60.8	59.7
SAPLMA	77.0	76.4	71.3	75.1	87.7	83.0	78.4	85.6	93.4	89.6	89.4	96.1	94.6	91.5
PRISM[1]	80.1	93.6	79.8	62.6	59.5	81.5	76.2	80.8	94.3	81.0	63.4	60.1	81.1	76.8
PRISM[2]	**84.4**	92.7	88.8	88.0	96.6	90.3	90.1	93.3	98.6	94.1	93.7	99.6	97.3	96.1
ReFL (ours)	83.6	**93.7**	**92.6**	**92.6**	**97.5**	**93.5**	**92.2**	**94.0**	**99.3**	**94.9**	**97.0**	**99.8**	**97.9**	**97.2**

Table 13: Overview of experimental results on the True-False dataset using Qwen2.5-14B. The table compares the ACC and AUC metrics (in %) for each semantic topic. PRISM[1] represents PRISM-MM, and PRISM[2] represents PRISM-SAPLMA.

used during training. The model is optimized with Adam using a learning rate of 1e-3, zero weight decay, and a dropout rate of 0.2, and is trained for 10 epochs. All experiments are repeated three times with different random initializations, and the final results are reported as the average over the three runs.

B.2 Extended Experimental Results on Diverse LLMs

To further demonstrate the robustness, generalizability, and broad applicability of our proposed ReFL framework, we conducted additional experiments on a wider range of open-source Large Language Models beyond those presented in the main paper. This appendix provides a comprehensive overview of ReFL's performance on these diverse models, including LLaMA2-13B, Qwen2.5-7B-instruct, and Qwen2.5-14B.

Table 14 summarizes the key details of all LLMs utilized in our study, encompassing both those featured in the main paper and these additional models. Subsequent tables detail the ACC and AUC metrics for ReFL and baseline methods on the True-False for these extended evaluations.

Table 11, Table 12 and Table 13 presents the

Model Name	Params	Type
LLaMA2-7B	7B	Base
LLaMA2-7B-Chat	7B	Instruction-tuned
LLaMA2-13B	13B	Base
LLaMA3-8B	8B	Base
OPT-6.7b	6.7B	Base
Qwen2.5-3B-Instruct	3B	Instruction-tuned
Qwen2.5-7B-Instruct	7B	Instruction-tuned
Qwen2.5-14B	14B	Base

Table 14: Details of All Large Language Models Used in Experiments

complete experimental results for ReFL and baseline methods on the True-False dataset using the LLaMA2-13B, Qwen2.5-7B-instruct and Qwen2.5-14B models. These results further support ReFL's strong performance across different LLM model scales.

C Additional Ablation Studies

C.1 Impact of Demonstration Number

In this part of the experiment, we analyze the effect of the number of in-context examples (k) on model performance. We set $k \in [0, 2, 4, 6, 8]$ and fix the correction ratio at 50%. As shown in Figure 2, when the number of examples is too small, the model receives insufficient semantic information, which limits its generalization ability. The prompt lacks diversity and contextual signals, making it harder for the model to identify hallucination-related features. Too many examples may introduce redundant information or even noise, which reduces the model's ability to learn from correction signals and increases computation cost.

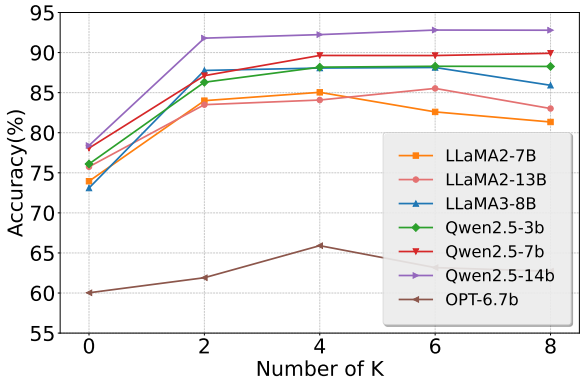

Figure 2: Effect of the number of in-context examples (k) on performance across various models.

C.2 Impact of Layer Selection

We investigated the effectiveness of using internal states from different layers of the LLM for classifier training and conducted a comprehensive comparison with the PRISM method. We selected multiple Transformer layers from the LLaMA2-7B model, including the last layer, the 28th, 24th, 20th, and 16th layers. For each layer, we extracted the hidden vector of the last token as the input feature. All other experimental settings remained unchanged, and the evaluation was performed on the True-False dataset.

As shown in Table 15, our method outperforms PRISM significantly, improving accuracy across all selected layers. As the selected layer moves closer to the input, the overall model performance tends to decline. The representation of the last layer is the most semantically abstract and closely aligned with the model output. These representations often contain the model's final judgment about the factuality of the input, shaped by both context and feedback. In contrast, shallower layers retain some syntactic or semantic features but often lack sufficient information or exhibit unstable generalization in classification tasks. These observations justify our design choice of using the last-layer representation in ReFL, as it provides the most informative and stable signal for hallucination detection.

C.3 Impact of Feature Extraction Strategies

The ReFL framework uses the embedding of the last token from the final layer of the model as the input feature for the classifier. To investigate whether this design choice is optimal, we conduct an ablation study comparing three feature extraction strategies. Specifically, we replace the original "last token embedding" with the representations obtained by applying mean pooling and max pooling overall token representations in the final layer. All other experimental settings are kept identical to ensure a fair comparison.

As shown in Table 16, both mean pooling and max pooling lead to a significant drop in performance. We hypothesize that mean pooling and max pooling introduce a large amount of irrelevant noise, which includes function words and irrelevant tokens that do not contribute to factuality judgment. In contrast, the last token of the model can be regarded as a compressed summary of the model's complete reasoning process, and therefore serves as the most discriminative feature for hallucination

Models	ReFL ACC (%)							PRISM ACC (%)						
	Anim.	Cities	Comp.	Elem.	Facts	Invent.	Avg.	Anim.	Cities	Comp.	Elem.	Facts	Invent.	Avg.
16th-layer	75.6	88.5	90.3	76.2	84.2	84.4	**83.2**	81.1	70.2	74.8	89.9	83.8	72.6	78.7
20th-layer	69.3	85.5	90.9	77.5	83.7	82.0	**81.5**	83.0	73.6	74.1	89.2	82.7	75.5	79.7
24th-layer	71.3	89.4	90.0	76.8	85.2	83.3	**82.6**	79.3	72.4	74.4	89.1	81.4	75.2	78.6
28th-layer	72.2	90.9	89.5	76.3	86.3	84.6	**83.3**	78.5	69.6	71.6	88.0	81.8	73.2	77.1
last-layer	72.6	93.6	90.6	78.1	88.3	87.2	**85.0**	71.6	72.7	71.4	84.6	80.4	75.8	76.1

Table 15: Experimental results on the True-False dataset using LLaMA2-7B, showing accuracy of ReFL and PRISM across different layers. ReFL outperforms PRISM in every layer, with the highest accuracy observed at the last-layer.

Methods	True-False ACC (%)							True-False AUC (%)						
	Anim.	Cities	Comp.	Elem.	Facts	Invent.	Average	Anim.	Cities	Comp.	Elem.	Facts	Invent.	Average
Average Pooling	52.7	58.5	50.9	56.9	66.9	59.9	57.6	69.0	69.4	62.2	59.2	75.9	76.9	68.8
Max Pooling	49.6	51.3	49.6	50.7	50.0	50.4	50.3	52.7	54.4	49.3	51.6	50.6	55.0	52.3
Last Token	**72.6**	**93.6**	**90.6**	**78.1**	**88.3**	**87.2**	**85.0**	**86.4**	**98.0**	**95.7**	**89.3**	**95.6**	**94.9**	**93.3**

Table 16: Overview of experimental results on the True-False dataset using LLaMA2-7B. The table compares the ACC and AUC metrics for Average Pooling, Max Pooling and Last Token methods.

detection.

D Direct Analysis of Internal Representations

To explicitly verify our core conceptual claim that the corrective triplets effectively act as structured positive and negative feedback to reshape the LLM's internal representation space, we conduct both a statistical quantification analysis and a visualization analysis of the internal states.

D.1 Statistical Analysis of Internal Representations

To accurately characterize the differences in structural salience and further validate the effectiveness of the ReFL framework, we conducted a statistical analysis of the internal representations. Specifically, we introduced the Variance Ratio as a direct metric to measure the structural salience along the "factuality direction." This ratio quantifies the proportion of total variance in the representation space that is aligned with the factuality direction (Marks and Tegmark, 2024; Zhang et al., 2025a). A higher Variance Ratio indicates that the model more actively focuses on distinguishing factual from hallucinated content.

We computed the corresponding variance ratios for the six subsets of the True-False dataset. As shown in Table 17, compared to the SAPLMA baseline, ReFL consistently enhances the variance ratio across almost all domains, achieving an average relative improvement of over 121%. This quantitative result demonstrates that ReFL's corrective feedback mechanism significantly enhances the salience of the internal factuality structure, effectively "reconstructing" the representation space to make factuality-related information more prominent.

Method	True-False						
	Anim.	Cities	Comp.	Elem.	Facts	Inv.	Avg.
SAPLMA	17.91	13.13	2.58	**13.77**	11.40	11.10	11.65
ReFL	**26.72**	**42.43**	**13.86**	13.71	**27.37**	**30.93**	**25.84**

Table 17: Experimental results of Variance Ratio on the True-False dataset, comparing the structural salience along the factuality direction between ReFL and SAPLMA baseline.

D.2 Visualization Analysis of Textual Representations

To further validate the effectiveness of the proposed ReFL framework for hallucination detection, we conducted a visualization analysis of the textual representations before classification. Specifically, we applied the t-distributed stochastic neighbor embedding (t-SNE) method to reduce the high dimensional embedding vectors and project them into a 2D space on the True-False dataset. This allowed us to visually examine how different categories are distributed in the low dimensional space. As shown in Figure 3, we present visualization results of our method (ReFL) and the baseline method (SAPLMA) across multiple factual topics, includ-

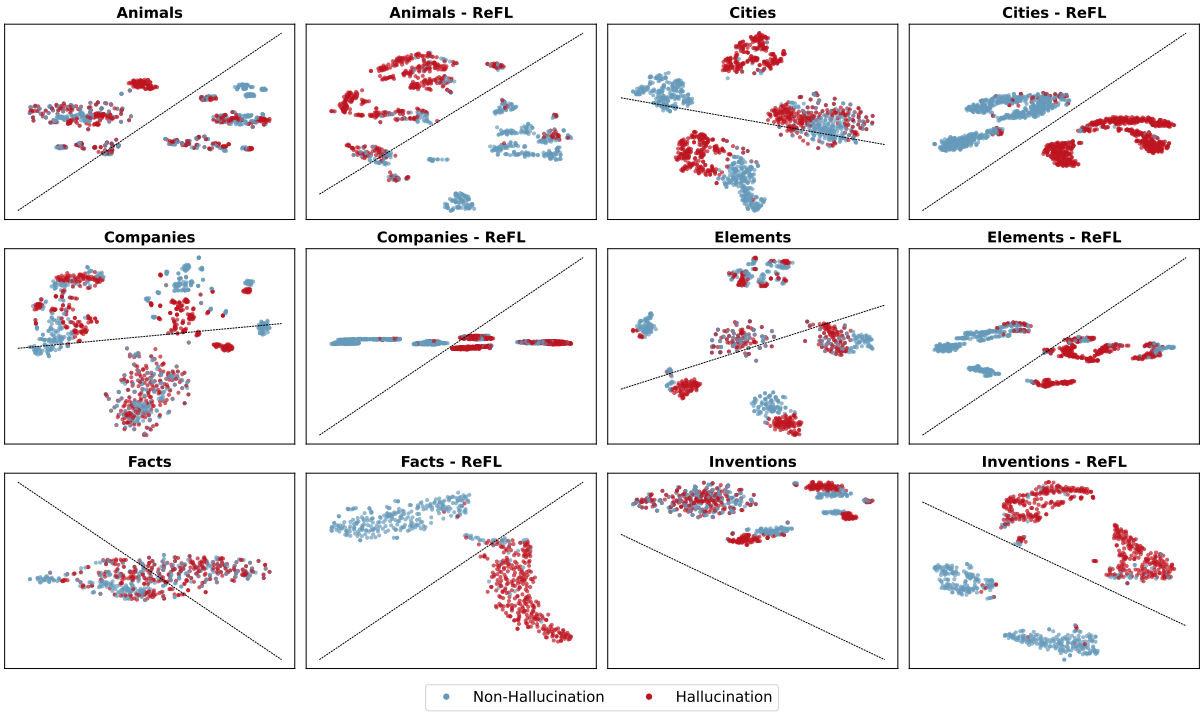

Figure 3: We apply t-SNE to visualize the internal representations of the LLaMA3-8B model on the True-False dataset, comparing results before and after the introduction of the ReFL framework. Red points indicate hallucinated samples, and blue points represent non-hallucinated ones. A logistic regression classifier is fitted to distinguish between the two classes, with its decision boundary shown as a black dashed line. The plot demonstrates that we enhances the separation between hallucinated and non-hallucinated samples.

ing Animals, Cities, Companies, Elements, Facts, and Inventions. In each subplot, red dots represent hallucinated samples and blue dots represent non-hallucinated samples. We also trained a logistic regression classifier in the t-SNE-reduced space and plotted its decision boundary (dashed line) to further assess class separability. This comparison provides a clear and intuitive view of the model's embedding structure and helps assess the alignment of its internal state with hallucination-related semantics.

From the Figure 3, we observe that the embeddings generated by ReFL exhibit clearer class boundaries and greater inter-class separation in the low-dimensional space. This indicates that we promote more structured and topic-consistent internal representations. For example, in the Facts and Inventions tasks, the embeddings from our method show more compact distributions and clearer boundaries between hallucinated and non-hallucinated samples, indicating stronger discriminative capability. Even for topics with greater semantic variation, such as Cities and Companies, we still achieve better class separability. These results suggest that our method not only improves the LLM's ability to represent factual semantics but also enhances its accuracy in distinguishing hallucinated from factual content.Compared to SAPLMA, ReFL produces representations that are more robust to topic shifts and more aligned with task relevant semantic boundaries.

Overall, this visualization analysis qualitatively supports our key hypothesis: by incorporating corrective context from prediction label pairs, the ReFL framework effectively guides structural adjustments of internal representations. This improves the model's ability to represent and separate hallucinated content in the embedding space, further confirming the effectiveness of our approach in semantic representation and hallucination detection.

E Detailed Probability Calculation

This appendix provides the detailed mathematical formulations for computing the predicted label probability, as referenced in the Methodology section of the main paper. These steps are standard in language-model-based classification and are included here for completeness and clarity.

To compute this probability, we follow the

token-level likelihood approach, commonly used in language-model-based classification. The model outputs unnormalized logits at each position in the sequence. These logits are transformed into a probability distribution over the vocabulary using the softmax function:

$$P(w_i) = \frac{\exp(\text{logit}_i)}{\sum_{j=1}^{V} \exp(\text{logit}_j)} \tag{6}$$

where V is the vocabulary size, w_i is the i-th token in the candidate label sequence, and logit_i is the LLM's output score for that token position. For each candidate label $y \in \mathcal{L}$, we append it as a suffix to the prompt (i.e., the combination of C_k and test input x) and pass the full sequence into the model. We then extract the model-predicted probabilities for each token in the label, and compute the joint likelihood of the entire label given the prompt:

$$P_M(y \mid C_k, x) = \prod_{i=1}^{n} P(w_i) \tag{7}$$

To eliminate the influence of label length on the computed probability, we apply geometric mean normalization, ensuring that longer labels do not unfairly receive lower scores purely due to their length:

$$\tilde{P}_M(y \mid C_k, x) = \left(\prod_{i=1}^{n} P(w_i) \right)^{1/n} \tag{8}$$

where $\tilde{P}_M$ denotes the normalized likelihood.

Finally, we normalize the geometric mean likelihoods of all labels to ensure their sum equals 1. The label with the highest normalized likelihood is selected as the model's final prediction.