Logical Consistency as a Bridge: Improving LLM Hallucination Detection via Label Constraint Modeling between Responses and Self-Judgments

Hao Mi **Qiang Sheng*** **Shaofei Wang** **Beizhe Hu** **Yifan Sun**
Zhengjia Wang **Hengqi Zeng** **Yang Li** **Danding Wang** **Juan Cao**
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
{mihao24s, shengqiang18z, caojuan}@ict.ac.cn

Abstract

Large Language Models (LLMs) are prone to factual hallucinations, risking their reliability in real-world applications. Existing hallucination detectors mainly extract micro-level intrinsic patterns for uncertainty quantification or elicit macro-level self-judgments through verbalized prompts. However, these methods address only a single facet of the hallucination, focusing either on implicit neural uncertainty or explicit symbolic reasoning, thereby treating these inherently coupled behaviors in isolation and failing to exploit their interdependence for a holistic view. In this paper, we propose **LaaB** (**L**ogical Consistency-**a**s-**a**-**B**ridge), a framework that bridges neural features and symbolic judgments for hallucination detection. LaaB introduces a "meta-judgment" process to map symbolic labels back into the feature space. By leveraging the inherent logical bridge where response and meta-judgment labels are either the same or opposite based on the self-judgment's semantics, LaaB aligns and integrates dual-view signals via mutual learning and enhances the hallucination detection. Extensive experiments on 4 public datasets, across 4 LLMs, against 8 baselines demonstrate the superiority of LaaB.[1]

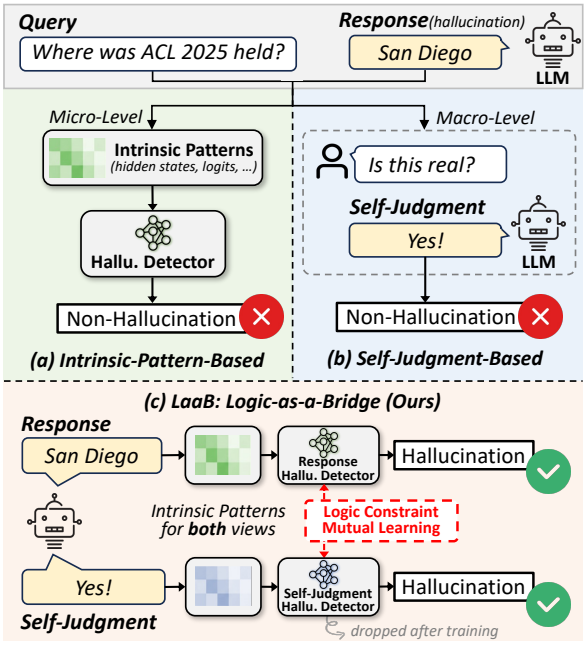

Figure 1: Comparison between existing hallucination detection methods (a-b) and our proposed LaaB (c). Unlike methods that rely solely on micro-level intrinsic patterns (a) or macro-level symbolic judgment (b), LaaB bridges these two views by enforcing logical consistency via logic-constraint mutual learning.

1 Introduction

Large language models (LLMs) have shown impressive capabilities across diverse tasks (Wei et al., 2022; Hu et al., 2024a; Zhao et al., 2026; Chen et al., 2026). However, their reliability in real-world applications is compromised by factual hallucinations: outputs that appear plausible but contradict verified facts or commonsense, even without malicious prompting (Huang et al., 2025). As recent studies indicate that hallucinations may be an inherent property of LLMs rather than a fully solvable error, their complete elimination remains elusive (Xu et al., 2025; Mohsin et al., 2026). Consequently, accurate hallucination detection is a critical requirement for maintaining reliable and trustworthy LLM-based systems (Ji et al., 2023a; Liu et al., 2024a; Zhang et al., 2025c; Hu et al., 2025).

Hallucination detection is generally formulated as a binary classification task that predicts the factuality of LLM responses. Besides applying fact-checking that relies on reliable external knowledge (Min et al., 2023), recent work looks *inside* LLMs by 1) extracting intrinsic patterns, or 2) using LLMs themselves as judges. The intrinsic-pattern-based detectors exploit the LLM's behavioral patterns during generation to quantify its un-

*Corresponding author.
[1]Our code and dataset are available at https://github.com/ICTMCG/LaaB

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6311–6330
July 2-7, 2026

certainty of the response (see Figure 1(a)), including generation consistency (Manakul et al., 2023; Farquhar et al., 2024), output confidence (Guo et al., 2017), hidden states (Azaria and Mitchell, 2023), and attention maps (Chuang et al., 2024), etc. These methods capture the nuanced internal signals from a micro perspective, but these metrics may lack proper calibration, resulting in the failure to identify high-certainty hallucinations (Tan et al., 2025; Wen et al., 2024; Zhou et al., 2024; Simhi et al., 2025). In contrast, the detectors based on self-judgments directly obtain LLMs' verbal judgments through factuality-oriented prompting (see Figure 1(b)), assuming that LLMs may elicit different knowledge when switching the role from answering to judging (Ji et al., 2023b; Li et al., 2025a). This paradigm exploits the macro-level judgment, but the verbal judgment suffers from self-preference bias (Wataoka et al., 2024; Panickssery et al., 2024) or overthinking issues (Zhang et al., 2025b; Su et al., 2025), possibly leading to "secondary" hallucination. This motivates us to explore: **How to effectively integrate the micro-level intrinsic signals and macro-level self-judgments for more accurate hallucination detection?**

To perform effective integration, we propose a hallucination detection method named **LaaB** (**L**ogical Consistency-**a**s-**a**-**B**ridge; see Figure 1(c)). The key challenge in the integration is to build a joint, learnable framework for both the neural features derived from intrinsic patterns and the symbolic judgments from the LLM itself. To address this issue, LaaB exploits the inherent logical constraint between the response and self-judgment as a bridge. First, we map the symbolic self-judgment back to the feature space to make it possible to optimize the hallucination detection via joint learning. Our idea starts from a simple but crucial fact: The LLM's self-judgment on its response is *also* a response that may be hallucinatory, and the intrinsic signal of the judgment generation may reveal its own veracity. By applying intrinsic-pattern-based methods on the self-judgment (*i.e.*, meta-judgment), LaaB obtains the learned features from the quantification of the self-judgment uncertainty. Subsequently, we transform the hallucination prediction of the self-judgment into that of the original response by leveraging the inherent logic constraint: the response and the self-judgment would share the same factuality label if the self-judgment claims that the response is truthful; otherwise, their labels should be opposite. Based on this logic rule, LaaB can obtain two aligned predictions for the original response from two neural modules of different views, thus enhancing the final judgment of hallucination. A mutual learning strategy is finally adopted for the whole optimization.

Our contributions are summarized as follows:

- **Concept:** We propose to see LLMs' self-judgment as a special response that can be checked by another hallucination detector, whose results will bridge the logic constraint that enables the integration of the predictions from both the response and self-judgment views.
- **Method:** We design LaaB, which bridges the prediction signals from both intrinsic patterns and self-judgments, and builds a mutual learning framework for accurate hallucination detection.
- **Performance:** Experiments on 4 public datasets, across 4 LLMs, against 8 baselines show that LaaB can effectively enhance the performance in hallucination detection without introducing significant additional inference cost.

2 Related Work

2.1 Hallucination Detection

Hallucination detection aims to evaluate the factuality of LLM outputs. Given that conventional fact-checking relies heavily on external knowledge retrieval and evidence verification (Hu et al., 2024b; Min et al., 2023; Wang et al., 2024; Wan et al., 2025; Chern et al., 2025), we focus on detection methods that leverage the LLM's internal signals, which fall into two categories: Intrinsic-pattern-based methods and self-judgment-based methods.

Intrinsic-pattern-based methods leverage the signals generated during the inference, positing that LLMs exhibit distinct internal behaviors when hallucinating compared to when generating factual content, typically including hidden states, prediction logits, and attention scores. *Hidden-state-based methods* assume that the truthful and hallucinated boundaries are encoded in the LLM's latent space, generally training classifiers on layer-wise hidden states (Li et al., 2023b; Azaria and Mitchell, 2023; Su et al., 2024; Du et al., 2024; Liu et al., 2024b; Park et al., 2025; Ni et al., 2025) or activation dynamics (Wang et al., 2025; Zhang et al., 2025d). *Logit-based methods* interpret the output probability distribution as a proxy for model confidence, where lower confidence or higher entropy often correlates with hallucination (Jiang et al., 2024;

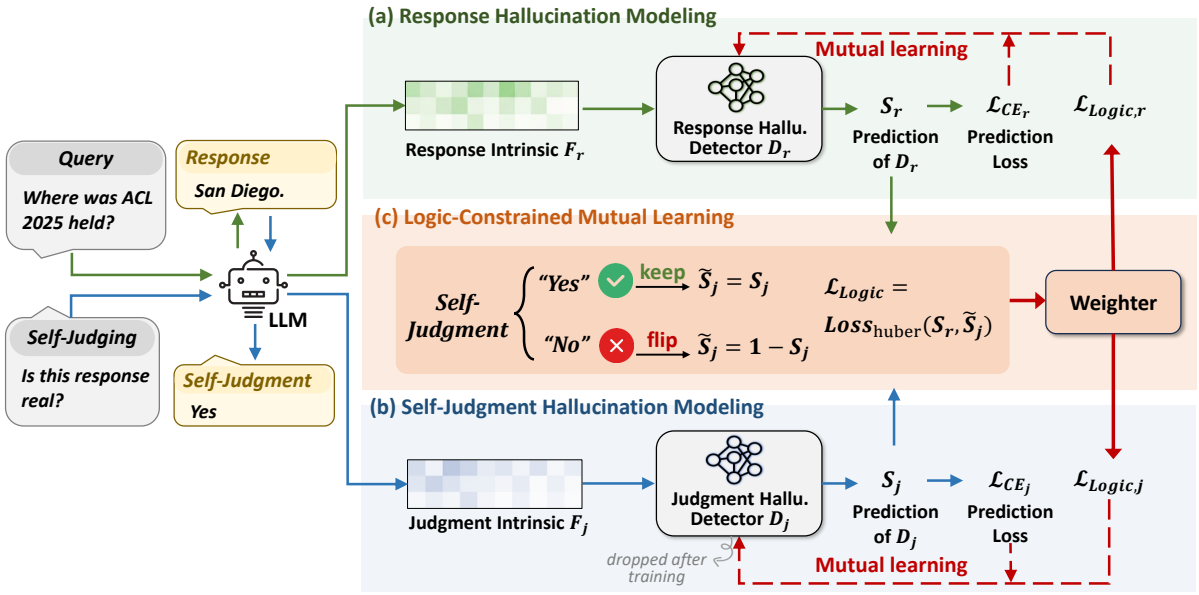

Figure 2: Overall architecture of **LaaB**. Given a user query and corresponding response, LaaB first performs **(a)** Response Hallucination Modeling, extracting intrinsic features from the response generation to capture implicit uncertainty. **(b)** Self-Judgment Hallucination Modeling introduces a meta-judgment process that analyzes the elicited verbal judgment to mitigate evaluative biases. Finally, **(c)** Logic-Constrained Mutual Learning bridges these dual views by enforcing logical consistency between response and judgment predictions, utilizing their inherent dependency for robust joint optimization.

Ma et al., 2025; Vashurin et al., 2025; Vazhentsev et al., 2025; Li et al., 2025b; Tan et al., 2026). *Attention-based methods* exploit the information flow, assuming hallucinations stem from improper attention to the context, and identify anomalies through attention map distributions (Chuang et al., 2024; Liu et al., 2025a; Binkowski et al., 2025; Qi et al., 2026). Despite capturing micro-level uncertainty, they often lack semantic calibration, failing to identify hallucinations with high confidence (Tan et al., 2025; Simhi et al., 2025; Li et al., 2025c).

Self-judgment-based methods leverage the semantic reasoning capabilities of LLMs to assess factuality through verbal interaction. Early works investigated the self-evaluation feasibility (Yin et al., 2023; Xiong et al., 2024), while subsequent research introduced mechanisms like self-correction (Ji et al., 2023b; Dhuliawala et al., 2024; Yuan et al., 2025; Zhang et al., 2026) and multi-agent debating (Liu et al., 2025b; Sun et al., 2025) to elicit accurate judgments. However, it inherently relies on the model's generation capabilities, making it susceptible to self-preference bias and "evaluative hallucination" (Wataoka et al., 2024; Zhang et al., 2025b). To address these limitations, we argue that intrinsic signals and verbal judgments are complementary: the former quantifies micro-level uncertainty, while the latter provides logical anchoring. In this work, we propose to bridge these two perspectives by viewing the self-judgment not as a ground truth, but as another generative behavior subject to hallucination. By modeling the logical constraint between the *response* and the *self-judgment*, we integrate them into a mutual learning framework to enhance detection accuracy.

2.2 Mutual Learning

Mutual learning is a paradigm in which peer networks learn from each other to improve performance and generalization. Zhang et al. (2018) introduced deep mutual learning, where multiple students are jointly trained to align their output distributions with peers. Subsequent works regard mutual learning as a process of online knowledge distillation where the peer predictions serve as a dynamic teacher, and develop variants by introducing ensemble learning (Guo et al., 2020; Tan and Liu, 2022) and contrastive learning (Yang et al., 2023). In this paper, we adopt mutual learning to bridge the individual learning of hallucination detection modules for the original response and self-judgment (which serve as the peers in our method).

3 Task Formulation

We formulate hallucination detection as a binary classification task. Given a user query Q_r and an

Table 1: Summary of notations in LaaB Framework

Symbol	Description
Response	
Q_r	User query
O_r	LLM response to Q_r
F_r	Intrinsic features for O_r
D_r	Detector over F_r
$L_r \in \{0, 1\}$	Ground-truth for O_r (1: factual)
$\hat{L}_r \in \{0, 1\}$	Predicted label for O_r
S_r	Predicted distribution from D_r
Self-Judgment	
Q_j	Evaluation prompt on O_r
$O_j \in \{\text{Yes, No}\}$	LLM response to Q_j (Verbal judge)
F_j	Intrinsic features for O_j
D_j	Detector over F_j
$L_j \in \{0, 1\}$	Ground-truth for O_j (Meta judge)
$\hat{L}_j \in \{0, 1\}$	Predicted label for O_j
S_j	Predicted distribution from D_j

LLM-generated response O_r, the primary goal is to determine the factuality label $L_r \in \{0, 1\}$, where $L_r = 1$ denotes a factual response and $L_r = 0$ denotes a hallucination response.

For an intrinsic-pattern-based detector D_r, it maps intrinsic model features F_r (derived during the generation process of O_r) to a predicted probability distribution S_r over the labels, and then gets the predicted label $\hat{L}_r$. For a self-judgment-based detector, it generates a verbal judgment $O_j \in \{\text{“Yes”}, \text{“No”}\}$ based on the original response O_r and a factuality evaluation prompt Q_j, where “Yes” and “No” can also correspond to the predicted label $\hat{L}_r = 1$ and 0, respectively. For the meta-judgment we will detail in § 4, similar to D_r, a secondary detector D_j maps the intrinsic features F_j for O_j to the ground-truth label of the judgment factuality $L_j \in \{0, 1\}$. The consolidated list of notations used in the LaaB framework is presented in Table 1.

4 Proposed Method: LaaB

Figure 2 overviews our proposed LaaB, which first models response hallucination and self-judgment hallucination, respectively, and then adopts logic-constrained mutual learning to guide the joint optimization. We detail the components' design below.

4.1 Response Hallucination Modeling

This module detects hallucinations using intrinsic patterns for generating O_r. We utilize an MLP-based detector D_r that accepts features F_r and outputs a probability distribution $S_r = (S_{r\_hallu}, S_{r\_real})$. The extracted features include:

Hidden States (H_r). Following Azaria and Mitchell (2023), we assume the hidden representation of final token in a sequence aggregates semantic information. We feed the concatenated sequence pair (Q_r, O_r) into the LLM and extract the hidden state of the last token at the validation-optimal layer $K_{\text{val},r}$, denoted as H_r.

Prediction Logits (P_r). We adopt the Logits Lens hypothesis (nostalgebraist, 2020; Jiang et al., 2024), which assumes that all hidden layers of the LLM share the same unembedding space with the output layer. Let P_{ll} represent the layer-wise probabilities of the generated tokens in O_r. We aggregate P_{ll} via a single Transformer layer followed by mean-pooling to obtain the sequence-level feature P_r.

Attention Scores (A_r). Following Chuang et al. (2024), we utilize the “Lookback” ratio, which quantifies the token-level attention assigned to preceding context components. Building upon this idea, we further adapt it to our context for factual hallucination detection. For each token in O_r, we compute attention ratios targeting four segments: the system prompt, query Q_r, response trigger, and preceding tokens of O_r. We pool these token-wise ratios and select the top-P informative heads based on KL divergence to form the feature A_r.

The details of feature extraction are provided in Appendix B. In the implementation, D_r uses one feature in $\{H_r, P_r, A_r\}$ as the input F_r and is trained via cross-entropy loss $\mathcal{L}_{\text{CE,r}}$ against L_r.

4.2 Self-Judgment Hallucination Modeling

We define *evaluative hallucination* as the scenario where an LLM acting as a judge produces an incorrect assessment O_j. Rather than accepting O_j as ground truth, we treat it as a generated artifact with a learnable factuality label L_j. We train a detector D_j to predict L_j based on intrinsic features F_j extracted during the generation of O_j. Similar to §4.1, we extract features $F_j \in \{H_j, P_j, A_j\}$, with specific adaptations for the judgment context:

Hidden States (H_j). The states are extracted from the last token of the judgment O_j at layer $K_{\text{val},j}$.

Prediction Logits (P_j). Based on Logits Lens, we first extract the layer-wise probability distribution of the first token of O_j. Then we construct semantic sets V_{yes} and V_{no} (grouping synonyms for “Yes”/“No”, Appendix B.2) and aggregate their probabilities into P_{yes} and P_{no}. To emphasize the

Table 2: Logic constraints between the truth value of the response factuality and self-judgment factuality based on the LLM self-judgment.

Self-Judgment (O_j)	**Interpretation**	**Target** (L_r)
"Yes"	**Affirmation**: The LLM supports the original response. Keep labels consistent.	L_j
"No"	**Negation**: The LLM refutes. Reverse the factuality.	$1 - L_j$

contrast, the input feature is constructed as:

$$P_j = \begin{cases} P_{\text{yes}} \oplus (P_{\text{yes}} - P_{\text{no}}), & \text{if } O_j = \text{"Yes"}, \\ P_{\text{no}} \oplus (P_{\text{no}} - P_{\text{yes}}), & \text{if } O_j = \text{"No"}. \end{cases} \tag{1}$$

Attention Scores (A_j). We segment the judgment context Q_j into six components (Framing, Query, Response, Eval_Query, Format, and Trigger) and compute the attention ratios of the judgment token over these segments to form A_j.

4.3 Logic-Constrained Mutual Learning

Logical Dependency. In the self-judgment setting, the verbal judgment O_j implies the factuality prediction of the response O_r, and we can derive the logical consistency between L_r and L_j. Specifically, when O_j is "Yes", the LLM predicts $\hat{L}_r = 1$. If $L_j = 1$, then the above prediction is correct and $L_r = 1$; otherwise, the prediction is incorrect and $L_r = 0$. Similarly, when O_j is "No", the relationship between the labels is reversed. Logical dependency is summarized in Table 2.

Framework. Detectors D_r and D_j produce probability distributions $S_r = (S_{r,\text{hallu}}, S_{r,\text{real}})$ and $S_j = (S_{j,\text{hallu}}, S_{j,\text{real}})$, respectively. To enforce logical consistency, we employ the Huber loss (Huber, 1964), denoted as $\mathcal{L}_{\text{Huber}}$, to align the scalar probabilities of the two detectors. The logic-constrained loss $\mathcal{L}_{\text{Logic}}$ is defined as:

$$\mathcal{L}_{\text{Logic}} = \begin{cases} \mathcal{L}_{\text{Huber}}(S_{r,\text{hallu}}, S_{j,\text{hallu}}), & \text{if } O_j = \text{"Yes"}, \\ \mathcal{L}_{\text{Huber}}(S_{r,\text{hallu}}, S_{j,\text{real}}), & \text{if } O_j = \text{"No"}. \end{cases} \tag{2}$$

This loss encourages D_r and D_j to learn robust representations by aligning their predictions according to the logical dependency. More details about Huber loss are provided in Appendix C.

To prevent mutual degradation (where a weaker detector misleads a stronger one), we introduce a confidence-aware weighting mechanism. We assume a detector with higher confidence on the ground truth label possesses better representations. For a sample pair, we compute a weight based on the ratio of the peer's confidence to the self's confidence. The specific logic losses for D_r and D_j are:

$$\mathcal{L}_{\text{Logic},r} = \log\left(1 + \frac{S_j(L_j)}{S_r(L_r)}\right) \cdot \mathcal{L}_{\text{Logic}}, \tag{3}$$

$$\mathcal{L}_{\text{Logic},j} = \log\left(1 + \frac{S_r(L_r)}{S_j(L_j)}\right) \cdot \mathcal{L}_{\text{Logic}}, \tag{4}$$

where $S(L)$ denotes the predicted probability of the ground truth class. Finally, the total training objective for each detector ($* \in \{r, j\}$) combines the cross-entropy loss and the weighted logic loss:

$$\mathcal{L}_* = \mathcal{L}_{\text{CE},*} + \alpha_* \mathcal{L}_{\text{Logic},*}. \tag{5}$$

Here, α_* is a batch-level balancing coefficient dynamically computed using the ratio of gradient norms with respect to the last-layer parameters θ_*^{-1}, ensuring stable optimization:

$$\alpha_* = \frac{\left\| \nabla_{\theta_*^{-1}} \mathcal{L}_{\text{CE},*} \right\|_2}{\left\| \nabla_{\theta_*^{-1}} \mathcal{L}_{\text{Logic},*} \right\|_2 + \epsilon}. \tag{6}$$

Training Strategy. We adopt a two-stage training strategy. In the first stage, D_r and D_j are trained asynchronously in a round-robin manner; when one converges, it is frozen while the other continues. In the second stage, both detectors are jointly fine-tuned using the combined loss:

$$\mathcal{L}_{\text{Joint}} = \mathcal{L}_{\text{CE},r} + \mathcal{L}_{\text{CE},j} + \alpha \mathcal{L}_{\text{Logic}}. \tag{7}$$

Inference. During inference, only D_r is deployed. This allows the system to benefit from the knowledge distilled from D_j via mutual learning, without incurring the additional computational costs associated with the judgment generation. The pseudocode for the LaaB training and inference procedures is provided in Appendix H.

5 Experiments

5.1 Experimental Settings

Datasets. We utilize four widely used datasets for factualness hallucination detection: TriviaQA, MMLU, NQ_Open, and HaluEval (Appendix A.1). For each dataset, we 1) prompt the LLM to generate responses to the given query; 2) prompt the

Table 3: Performance comparison of baselines and LaaB in hallucination detection. **Bolded** numbers denote that the use of LaaB is better-performing than its corresponding base version. Underlined numbers are the highest in each column within each LLM group.

LLM	Method	TriviaQA		MMLU		NQ_Open		HaluEval		Average	
		macF1	Acc	macF1	Acc	macF1	Acc	macF1	Acc	macF1	Acc
Llama-3.1-8B-Instruct	Self-Judge	67.67	71.36	57.29	65.13	59.12	59.14	60.89	62.78	61.24	64.60
	SelfCheckGPT	76.34	81.98	59.78	63.56	69.77	73.17	74.41	74.49	70.08	73.30
	Eigen-Score	69.92	75.58	57.88	63.47	64.82	67.07	66.34	66.81	64.74	68.23
	SAPLMA	77.05	79.87	69.96	70.75	73.01	75.15	75.70	75.88	73.93	75.41
	+*LaaB*	**78.74**	**82.09**	**71.77**	**73.25**	**73.10**	**77.13**	**77.20**	**77.60**	**75.20**	**77.52**
	Logits Lens	72.50	74.81	65.27	66.07	62.63	63.26	72.21	72.48	68.15	69.16
	+*LaaB*	**75.11**	**78.88**	**66.72**	**70.65**	**65.72**	**68.90**	**72.29**	**73.04**	**69.96**	**72.87**
	Lookback Lens	71.65	74.45	72.74	73.96	68.60	70.88	75.13	75.49	72.03	73.70
	+*LaaB*	**73.58**	**77.44**	70.96	72.50	**72.63**	**75.46**	**77.00**	**77.54**	**73.54**	**75.74**
Llama-3.1-70B-Instruct	Self-Judge	66.00	83.58	64.30	82.18	68.65	79.21	65.27	66.82	66.06	77.95
	SelfCheckGPT	72.15	86.12	50.75	69.70	67.73	77.31	75.57	76.11	66.55	77.31
	Eigen-Score	66.97	82.37	59.39	77.48	60.59	71.60	64.41	64.43	62.84	73.97
	SAPLMA	71.12	83.13	71.19	79.62	71.64	78.62	77.17	77.22	72.78	79.65
	+*LaaB*	**73.20**	**86.22**	**71.40**	**81.75**	**71.71**	78.48	**79.15**	**79.44**	**73.87**	**81.47**
	Logits Lens	65.67	76.55	62.42	70.63	54.50	60.03	72.33	72.52	63.73	69.93
	+*LaaB*	**70.23**	**84.19**	60.18	**72.33**	**57.21**	**71.74**	72.07	72.36	**64.92**	**75.16**
	Lookback Lens	69.97	79.53	68.95	75.07	64.30	68.67	78.31	78.54	70.38	75.45
	+*LaaB*	**72.45**	**85.11**	**69.73**	**79.62**	**66.65**	**76.28**	**78.73**	**78.91**	**71.89**	**79.98**
Qwen-2.5-32B-Instruct	Self-Judge	72.19	78.73	65.14	77.54	71.29	73.99	69.07	69.28	69.42	74.89
	SelfCheckGPT	72.83	79.73	52.17	76.86	74.83	76.81	74.17	74.24	68.50	76.91
	Eigen-Score	67.52	73.51	56.98	75.35	61.33	62.63	69.19	69.24	63.76	70.18
	SAPLMA	76.73	80.09	69.38	78.38	76.81	77.55	76.65	76.76	74.89	78.20
	+*LaaB*	**79.42**	**83.85**	**69.60**	**78.99**	**79.35**	**80.65**	**77.17**	**77.19**	**76.39**	**80.17**
	Logits Lens	68.15	72.74	53.19	63.17	65.02	66.03	72.56	72.62	64.73	68.64
	+*LaaB*	**70.90**	**77.06**	**54.62**	**74.36**	**67.21**	**70.61**	**73.80**	**73.91**	**66.63**	**73.99**
	Lookback Lens	76.59	79.99	68.14	75.07	76.62	77.99	78.46	78.48	74.95	77.88
	+*LaaB*	**77.66**	**82.36**	**70.29**	**79.79**	**78.08**	**79.32**	78.16	78.21	**76.05**	**79.92**
Mistral-7B-Instruct-v0.3	Self-Judge	56.84	73.55	42.28	61.80	46.86	64.81	43.10	49.54	47.27	62.43
	SelfCheckGPT	68.67	75.69	56.73	60.40	66.31	68.07	72.60	72.61	66.08	69.19
	Eigen-Score	67.78	72.06	55.84	59.33	59.14	62.67	65.11	65.35	61.97	64.85
	SAPLMA	76.72	79.02	70.89	71.36	72.96	74.36	74.87	75.00	73.86	74.94
	+*LaaB*	**77.69**	**80.40**	**70.99**	71.26	**74.48**	**75.11**	**76.04**	**76.46**	**74.80**	**75.81**
	Logits Lens	68.72	72.52	59.56	60.26	58.94	62.82	68.53	68.87	63.94	66.12
	+*LaaB*	67.73	**73.54**	**60.98**	**62.91**	58.94	62.07	68.26	68.66	**63.98**	**66.80**
	Lookback Lens	73.81	75.95	69.61	69.68	71.46	72.26	74.82	75.00	72.43	73.22
	+*LaaB*	**74.46**	**77.02**	**71.24**	**72.10**	**71.67**	**73.01**	74.32	**75.11**	**72.92**	**74.31**

LLM to self-evaluate its responses using the template (Appendix A.2); 3) annotate the factuality of responses and judgments based on the ground-truth (Appendix A.3); and 4) split the resulting dataset with a 7:1:2 ratio for the training, validation, and testing sets.

Large Language Models. We use four commonly-used open-source LLMs, including Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct, Qwen-2.5-32B-Instruct, and Mistral-7B-Instruct-v0.3, which cover different families and scales (Appendix D).

Baselines. We include self-judgment-based detector Self-Judge (Kadavath et al., 2022) and five intrinsic-pattern-based detectors. The trainable detectors using internal model representations are detailed in §4.1, including SAPLMA (with hidden states), Logits Lens (with logits), and Lookback Lens (with attention patterns). The latter two include consistency-based methods, SelfCheckGPT (Manakul et al., 2023) and EigenScore (Chen et al., 2024), which are detailed in Appendix F.

Evaluation Metrics. Macro F1 (macF1) and accuracy (Acc) are adopted as evaluation metrics for hallucination detection, reflecting class-level and instance-level performance, respectively.

Implementation Details. For representation-based detectors, we configure the architectures of D_r and D_j as follows. For SAPLMA and Lookback Lens, both D_r and D_j employ a Multi-Layer Perceptron (MLP) classifier with hidden dimensions of $[256, 128, 64]$. For Logits Lens, we use a single Transformer layer with 4 attention heads to aggregate token-level features, followed by a MLP D_r configured with $[64, 16]$ for classification; while

D_j utilizes an MLP with hidden dimensions of $[128, 64, 16]$. We select the optimal learning rate from [1e-4, 5e-4, 1e-3, 5e-3] based on validation performance to conduct asynchronous training of D_r and D_j (the same lr selection criterion for baseline), followed by the joint fine-tuning with a learning rate of 1e-6. All classifiers are optimized using AdamW with a weight decay of 1e-5 and a dropout rate of 0.1. We adopt an early stopping strategy based on validation loss with a patience of 10 epochs.

For consistency-based detectors, each data instance is sampled 15 times using a decoding configuration of temperature=0.7, Top-p=0.9, and Top-k=10. Owing to the training-free attribute, we determine the optimal classification threshold via a grid search (step size = 0.1) on the validation set and report the results obtained on the test set.

5.2 Main Results

To evaluate the effectiveness of the **LaaB** framework, we conduct extensive experiments on four LLMs across four benchmarks. By comparing LaaB-enhanced detectors against various baselines, we aim to assess their capability in mining hallucination patterns by integrating information in two views guided by the logical constraints. The main results are summarized in Table 3. We observe that:

1) LaaB yields additional performance gains for the baselines using modeling intrinsic patterns in most cases. The sustained improvement observed across most settings demonstrates the method's robust adaptability along two key dimensions. *First*, LaaB exhibits exceptional generalization across diverse intrinsic patterns. As indicated by the **bold** values, it consistently augments detection capabilities when applied to hidden states (SAPLMA), logits (Logits Lens), and attention patterns (Lookback Lens). *Second*, LaaB maintains its efficacy across varying model scales (ranging from 7B to 70B) and distinct architectures (including LLaMA, Qwen, and Mistral), underscoring its generalizability in capturing fundamental hallucination signatures across different LLM families.

2) The hidden-state-based method SAPLMA shows higher detection performance than others and benefits more from LaaB. Among base detectors, SAPLMA achieves higher detection performance than logits-based and attention-based methods in most cases. With the LaaB enhancement, SAPLMA shows more stable improvements and predominantly attains the best results across the majority of evaluation settings. This performance gap suggests that denser hidden-state representations possess a higher capacity for retaining informative signals, which are critical for effective hallucination detection. In contrast, logits-based detectors (Logits Lens) are limited by their inherent sparsity and discreteness, thus perform relatively worse among the three intrinsic pattern types.

3) Detectors leveraging intrinsic patterns derived from LLMs' internal states outperform those leveraging the self-judgments or sampling estimates. Detectors like SAPLMA, Logits Lens, and Lookback Lens show higher accuracy and macro F1 scores than Self-Judge and sampling-based baselines in most cases. This observation suggests that intrinsic model representations encode richer factuality-related signals than discrete token-level outputs, and based on that, the mutual learning from LaaB provides a more effective mechanism for exploiting such signals.

5.3 Variant Analysis

We evaluate two variants of LaaB to dissect the contributions of response and judgment hallucination detectors. **1)** *LaaB (D_j)* uses only the judgment detector D_j at inference to predict the self-judgment label and then derives the response-level decision according to the logical dependency shown in Table 2. **2)** *LaaB ($D_j + D_j$)* utilizes both the response and judgment detectors at inference and averages their scores under the logical constraints.

The results are shown in Table 4, LaaB mostly outperforms LaaB (D_j), suggesting that the intrinsic patterns of responses remain the most informative cues for hallucination detection. D_j primarily serves to complement D_r by supplying additional signals from the self-judgment perspective. Moreover, LaaB ($D_j + D_j$) provides only marginal gains over LaaB despite nearly doubling LLM inference cost, indicating that most benefits of the judgment view are already distilled into the response detector D_r during training. Overall, these ablations suggest that logic-constrained mutual learning successfully integrates both perspectives into the response detector, enabling efficient and effective hallucination detection using only D_r at inference time.

5.4 Cross-Dataset Generalization

To assess the generalizability of our method under dataset shift, we adopt a leave-one-dataset-out evaluation protocol over four datasets. For each held-out benchmark, the hallucination detector is

Table 4: Performance comparison of LaaB and its variants on Llama-3.1-8B-Instruct

Method	Variant	TriviaQA		MMLU		NQ_Open		HaluEval	
		macF1	Acc	macF1	Acc	macF1	Acc	macF1	Acc
SAPLMA	LaaB	78.74	82.09	71.77	73.25	73.10	77.13	77.20	77.60
	LaaB (D_j)	75.71	81.26	69.23	71.50	71.72	77.29	77.36	77.99
	LaaB ($D_r + D_j$)	79.21	83.12	71.70	73.39	74.16	78.51	77.92	78.43
Logits Lens	LaaB	75.11	78.88	66.72	70.65	65.72	68.90	72.29	73.04
	LaaB (D_j)	45.17	72.64	62.57	69.28	41.02	67.07	70.91	71.65
	LaaB ($D_r + D_j$)	73.97	81.72	63.00	69.66	67.89	76.07	75.24	75.76
Lookback Lens	LaaB	73.58	77.44	70.96	72.50	72.63	75.46	77.00	77.54
	LaaB (D_j)	69.42	78.63	61.89	69.19	67.87	75.61	71.99	73.10
	LaaB ($D_r + D_j$)	75.26	80.74	63.69	69.90	72.24	77.44	76.61	77.43

Table 5: Leave-one-out Cross-Dataset Generalization Performance (macF1) on Llama-3.1-8B-Instruct. Both the baselines and LaaB are trained on the remaining three datasets and evaluated on the held-out benchmark.

Method	TriviaQA	MMLU	NQ_Open	HaluEval
SAPLMA	73.90	56.19	65.76	67.49
+*LaaB*	**78.18**	**59.31**	**68.33**	**70.25**
Logits Lens	73.05	56.65	57.25	67.78
+*LaaB*	**73.13**	55.05	**61.06**	**68.89**
Lookback Lens	74.09	60.05	65.44	68.25
+*LaaB*	72.61	**63.61**	**68.00**	**69.92**

trained on the other three datasets and then evaluated on the held-out benchmark.

As shown in Table 5, equipping diverse hallucination detectors with LaaB consistently improves cross-dataset performance in most cases, suggesting enhanced robustness to distribution shift. One plausible explanation is that our logic-constrained mutual learning bridges two complementary signals—intrinsic prediction representations and self-judgment patterns. By encouraging logical agreement across two views, the training objective discourages reliance on spurious, dataset-specific cues that are unlikely to be supported by both signals, thereby pushing the detector toward evidence that transfers across datasets. Consequently, the learned decision boundary appears less sensitive to dataset-specific characteristics and transfers more reliably to unseen benchmarks.

5.5 Further Analysis

We perform further analysis to find out how LaaB brings detection improvement by breaking down the test set into subsets from different views. Specifically, we focus on the origin of the corrected instances and the length distribution. The following analysis is based on experiments with Llama-3.1-8B-Instruct, aggregated as the average across

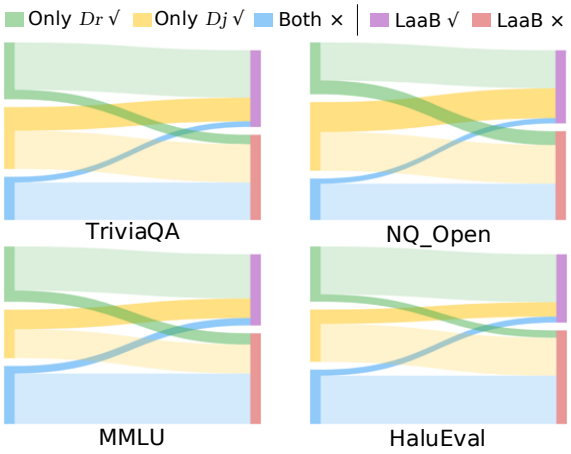

Figure 3: Prediction correctness transitions before and after applying LaaB.

SAPLMA, Logits Lens, and Lookback Lens.

How do the predictions change before and after applying LaaB? We categorized all testing data into four groups according to the independently trained detectors D_r and D_j: Only D_r ✓(), only D_j ✓(), both × (), and both ✓. For the categories except "both ✓", we visualize the prediction correctness transition with the sankey diagram (Figure 3), where flow widths are proportional to the absolute instance counts. We see that:

1) After adopting our proposed LaaB, a substantial portion of samples that D_r originally predicted wrongly were corrected (the yellow flow to purple end), indicating that LaaB indeed transfers knowledge for the detection of hallucination from the self-judgment view to the response view.

2) Compared to the loss that D_r turns into incorrect predictions (the green flow to red end), LaaB preserves most correct predictions of D_r.

3) Interestingly, we find a small but consistent flow from the "both ×" category to the correct class on all four datasets (the blue flow to purple end),

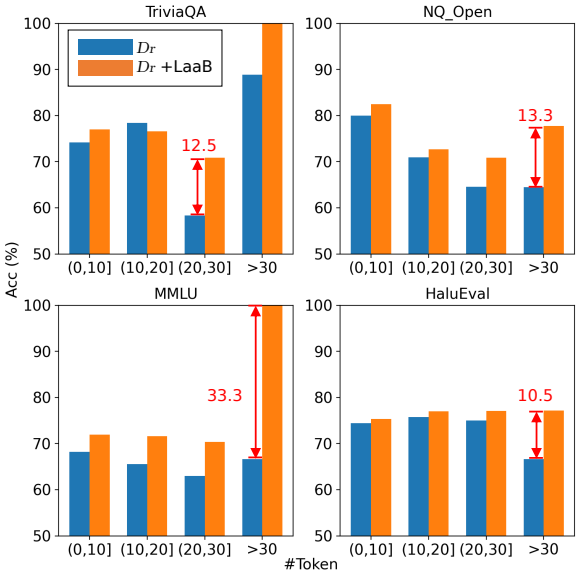

Figure 4: Performance on testing subsets with instances in different length intervals.

which is beyond expectation. This might be because the logical constraints from LaaB introduced weak yet useful supervision signals that allow D_r to refine the learned representations, even when both D_r and D_j predicted incorrectly before.

How effective is LaaB for instances with varying lengths? We calculate the accuracy of the subsets containing test instances in different length intervals. As presented in Figure 4, LaaB brings accuracy improvement for most length intervals on all four datasets, demonstrating its general applicability. Among the subsets, we see a greater improvement at the long text intervals like $(20, 30]$ and > 30. And the largest improvement occurs for the instances longer than 30 tokens at the MMLU dataset. This indicates that the hallucination detector can better handle more complex responses than before under the LaaB training. Inspired by Zhang et al. (2025a), a possible explanation is that self-judgment compresses response-level factuality into a single token ("Yes" or "No"), thereby mitigating representation sparsity and noise issues that arise with increasing sequence length.

6 Conclusion

We introduced LaaB (Logical Consistency-as-a-Bridge), a framework for LLM hallucination detection that bridges the gap between micro-level intrinsic neural patterns and macro-level symbolic self-judgments. By introducing a "meta-judgment" process to map symbolic labels back into the feature space, LaaB provides a mutual learning framework between the hallucination detection modules for the response and the LLM self-judgment, guided by the inherent logical constraint. Extensive experiments on four public benchmarks, across four open-source LLMs, against eight baseline models show that LaaB can improve the hallucination detection performance for most base models, without introducing a significant increase in the inference cost.

Future Work. We plan to further explore the following directions: 1) Evaluate LaaB and develop an improved version for hallucination detection in long articles; 2) Extend the core design to the multi-modality scenarios; and 3) Build a unified framework that integrates methods like LaaB with fact-checking pipelines.

Acknowledgment

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (XDB0680202), the National Natural Science Foundation of China (62406310), the China Postdoctoral Science Foundation (2024M763336), the Innovation Funding of ICT, CAS (E561160), and the Postdoctoral Fellowship Program of CPSF (GZC20232738, YJB20250186).

We adhere to the ACL Policy on Publication Ethics, including its Guidelines for Generative Assistance in Authorship (Cahill et al., 2025). We used generative AI tools solely for language polishing of the manuscript and for assistance in drafting certain portions of the code. All generated content was reviewed and verified by the authors.

Limitations

In this paper, we propose the integrated framework LaaB to combine the signals from LLMs' intrinsic patterns and self-judgments for hallucination detection. Despite its effectiveness, we identify the following limitations:

1) To obtain the self-judgment from the LLMs, we force the LLM to answer with "Yes" or "No", which is to avoid the situation where the LLM is originally intended to respond with "I don't know." The compulsory constraint of the candidate answer space may bring some noise that negatively influences detectors' training.

2) Theoretically, our integration cannot correct the result of the samples that both intrinsic-pattern-based and self-judgment-based methods make incorrect predictions. The joint learning procedure

may help some samples of this kind, but such an effect should be attributed to the joint optimization against ground-truth labels. Furthermore, our method does not guarantee the complete logical consistency for each sample in the resulting model because the applied constraint is soft.

3) The used intrinsic patterns and self-judgments are derived from the LLM that generates the given response, so our method is only applicable for the LLM service provider to monitor its service. The method cannot be used for third-party users who cannot access the internal information of LLMs. For these cases, fact-based misinformation detection (Sheng et al., 2021) or black-box hallucination detection (Bai et al., 2026) methods are more suitable.

Ethical Considerations

Risks. Our work aims at detecting hallucinated LLM outputs and is suitable to be a monitoring component for LLM services, thus reducing the risk of users being misled. Given that an individual detector could hardly be perfect, real-world applications should consider multiple ways to detect hallucination more accurately.

Data. This work uses four publicly released datasets, including TriviaQA, MMLU, NQ_Open, and HaluEval, under Apache-2.0 license, MIT license, Apache-2.0 license, and MIT license, respectively. We follow their intended use of academic research. During the research, we did not collect and use any unauthorized personal private data and did not recruit any human annotators.

References

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. Association for Computational Linguistics.

Yuzhuo Bai, Shuzheng Si, Kangyang Luo, Qingyi Wang, Wenhao Li, Gang Chen, Fanchao Qi, and Maosong Sun. 2026. Infi-check: Interpretable and fine-grained fact-checking of llms. *Preprint*, arXiv:2601.06666.

Jakub Binkowski, Denis Janiak, Albert Sawczyn, Bogdan Gabrys, and Tomasz Jan Kajdanowicz. 2025. Hallucination detection in LLMs using spectral features of attention maps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24354–24385. Association for Computational Linguistics.

Aoife Cahill, Leon Derczynski, and Kokil Jaidka. 2025. ACL Policy on Publication Ethics. Accessed: 2026-01-02.

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*.

Junjie Chen, Haitao Li, Minghao Qin, Yujia Zhou, Yanxue Ren, Wuyue Wang, Yiqun Liu, Yueyue Wu, and Qingyao Ai. 2026. Simulating dispute mediation with llm-based agents for legal research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29368–29375.

Ethan Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2025. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. In *Second Conference on Language Modeling*.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578. Association for Computational Linguistics.

Xuefeng Du, Chaowei Xiao, and Yixuan Li. 2024. Haloscope: Harnessing unlabeled llm generations for hallucination detection. *Advances in Neural Information Processing Systems*, 37:102948–102972.

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

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11017–11026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Beizhe Hu, Qiang Sheng, Juan Cao, Yang Li, and Danding Wang. 2025. Llm-generated fake news induces truth decay in news ecosystem: A case study on neural news recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 435–445. Association for Computing Machinery.

Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. 2024a. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 22105–22113.

Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024b. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Peter J Huber. 1964. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.

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

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. 2024. On large language models' hallucination with regard to known facts. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1041–1053. Association for Computational Linguistics.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

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

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Miaoran Li, Jiangning Chen, Minghua Xu, and Xiaolong Wang. 2025a. Hallucination detection in structured query generation via LLM self-debating. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16102–16113. Association for Computational Linguistics.

Rui Li, Jing Long, Muge Qi, Heming Xia, Lei Sha, Peiyi Wang, and Zhifang Sui. 2025b. Towards harmonized uncertainty estimation for large language models. In

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22938–22953. Association for Computational Linguistics.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. 2025c. Conftuner: Training large language models to express their confidence verbally. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Aiwei Liu, Qiang Sheng, and Xuming Hu. 2024a. Preventing and detecting misinformation generated by large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 3001–3004. Association for Computing Machinery.

Junteng Liu, Shiqi Chen, Yu Cheng, and Junxian He. 2024b. On the universal truthfulness hyperplane inside LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18199–18224. Association for Computational Linguistics.

Qiang Liu, Xinlong Chen, Yue Ding, Bowen Song, Weiqiang Wang, Shu Wu, and Liang Wang. 2025a. Attention-guided self-reflection for zero-shot hallucination detection in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21005–21021. Association for Computational Linguistics.

Zihang Liu, Jiawei Guo, Hao Zhang, Hongyang Chen, Jiajun Bu, and Haishuai Wang. 2025b. Long-form hallucination detection with self-elicitation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4082–4100. Association for Computational Linguistics.

Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with evidence. *Preprint*, arXiv:2502.00290.

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. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics.

Muhammad Ahmed Mohsin, Muhammad Umer, Ahsan Bilal, Zeeshan Memon, Muhammad Ibtsaam Qadir, Sagnik Bhattacharya, Hassan Rizwan, Abhiram R. Gorle, Maahe Zehra Kazmi, Nukhba Amir, Ali Subhan, Muhammad Usman Rafique, Zihao He, Pulkit Mehta, Muhammad Ali Jamshed, and John M. Cioffi. 2026. On the fundamental limits of llms at scale. *Preprint*, arXiv:2511.12869.

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 2025. Towards fully exploiting LLM internal states to enhance knowledge boundary perception. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24315–24329. Association for Computational Linguistics.

nostalgebraist. 2020. interpreting GPT: the logit lens. Accessed: 2026-01-02.

Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, and Yixuan Li. 2025. Steer LLM latents for hallucination detection. In *Forty-second International Conference on Machine Learning*.

Siya Qi, Yudong Chen, Runcong Zhao, Qinglin Zhu, Zhanghao Hu, Wei Liu, Yulan He, Zheng Yuan, and Lin Gui. 2026. Detecting contextual hallucinations in llms with frequency-aware attention. *Preprint*, arXiv:2602.18145.

Qiang Sheng, Xueyao Zhang, Juan Cao, and Lei Zhong. 2021. Integrating pattern-and fact-based fake news detection via model preference learning. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1640–1650.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: improving black-box ai-generated text detection via proxy-guided efficient re-sampling. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 494–502.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, I'm wrong: LLMs hallucinate with certainty despite knowing the answer. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14665–14688. Association for Computational Linguistics.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. 2025. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *Preprint*, arXiv:2505.00127.

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. Association for Computational Linguistics.

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2025. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE.

Chao Tan and Jie Liu. 2022. Online knowledge distillation with elastic peer. *Information Sciences*, 583:1–13.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. 2025. Too consistent to detect: A study of self-consistent errors in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4755–4765. Association for Computational Linguistics.

Hexiang Tan, Wanli Yang, Junwei Zhang, Xin Chen, Rui Tang, Du Su, Jingang Wang, Yuanzhuo Wang, Fei Sun, and Xueqi Cheng. 2026. Basecal: Unsupervised confidence calibration via base model signals. *Preprint*, arXiv:2601.03042.

Roman Vashurin, Maiya Goloburda, Preslav Nakov, and Maxim Panov. 2025. UNCERTAINTY-LINE: Length-invariant estimation of uncertainty for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7881–7908. Association for Computational Linguistics.

Artem Vazhentsev, Ekaterina Fadeeva, Rui Xing, Gleb Kuzmin, Ivan Lazichny, Alexander Panchenko, Preslav Nakov, Timothy Baldwin, Maxim Panov, and Artem Shelmanov. 2025. Unconditional truthfulness: Learning unconditional uncertainty of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35673–35694. Association for Computational Linguistics.

Yingjia Wan, Haochen Tan, Xiao Zhu, Xinyu Zhou, Zhiwei Li, Qingsong Lv, Changxuan Sun, Jiaqi Zeng, Yi Xu, Jianqiao Lu, Yinhong Liu, and Zhijiang Guo. 2025. FaStFact: Faster, stronger long-form factuality evaluations in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23814–23854. Association for Computational Linguistics.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156. Association for Computational Linguistics.

Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, and Rui Wang. 2025. Latent space chain-of-embedding enables output-free LLM self-evaluation. In *The Thirteenth International Conference on Learning Representations*.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230. Association for Computational Linguistics.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in LLM-as-a-judge. In *Neurips Safe Generative AI Workshop 2024*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Bingbing Wen, Chenjun Xu, Bin HAN, Robert Wolfe, Lucy Lu Wang, and Bill Howe. 2024. Mitigating overconfidence in large language models: A behavioral lens on confidence estimation and calibration. In *NeurIPS 2024 Workshop on Behavioral Machine Learning*.

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*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

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, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Chuanguang Yang, Zhulin An, Helong Zhou, Fuzhen Zhuang, Yongjun Xu, and Qian Zhang. 2023. Online knowledge distillation via mutual contrastive learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10212–10227.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665. Association for Computational Linguistics.

Yige Yuan, Bingbing Xu, Hexiang Tan, Fei Sun, Teng Xiao, Wei Li, Huawei Shen, and Xueqi Cheng. 2025. Fact-level calibration and correction for long-form

generations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 2807–2811. Association for Computing Machinery.

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. Association for Computational Linguistics.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. 2025b. Understanding the dark side of LLMs' intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27066–27101. Association for Computational Linguistics.

Wei Zhang, Guojun Dai, Ding Luo, Yan Wang, and Chen Ye. 2026. From hallucination to certainty: Meta-knowledge guided self-correcting large language models. *ACM Transactions on Intelligent Systems and Technology*.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4328.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025c. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. 2025d. ICR probe: Tracking hidden state dynamics for reliable hallucination detection in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17986–18002. Association for Computational Linguistics.

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, Xin Sun, Ya Zhang, Yongguo Yu, Kun Sun, and Weidi Xie. 2026. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, 651:775–784.

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3623–3643. Association for Computational Linguistics.

A Additional Details of Datasets

A.1 Introduction of Used Datasets

The detailed introduction of the four datasets used is as follows:

- **TriviaQA (Joshi et al., 2017):** A large-scale dataset for reading comprehension and question answering, consisting of trivia questions authored by enthusiasts and their associated evidence documents. We utilize the deduplicated validation split *(rc.nocontext subset)* with 9,961 Question-Answer Pairs.
- **MMLU (Hendrycks et al., 2021):** A benchmark covering 57 tasks across diverse domains, including STEM, humanities, social sciences, and professional knowledge. It evaluates a model's multitasking and general reasoning abilities using multiple-choice questions drawn from publicly available exams and academic sources. We utilize the test set with 14,041 multiple-choice questions.
- **NQ_Open (Kwiatkowski et al., 2019):** An open-domain question answering benchmark derived from Natural Questions. It contains real user queries paired with relevant passages from Wikipedia, designed to evaluate models' ability to retrieve and answer factual questions from large text corpora. We use its validation split without relevant passages, which contains 3,610 QA pairs.
- **HaluEval (Li et al., 2023a):** A large collection of generated and human-annotated hallucinated samples for evaluating the performance of LLMs in recognizing hallucination, including four subsets: QA, Dialogue, Summarization, and General Data. In our experiment, we use the QA split with 10K data samples.

A.2 Prompts for Response and Self-judgment

In this paper, we design the following simple prompt template to instruct the LLM to generate the response O_r:

Prompt for Response Generation

Answer the question concisely:
Q: $\{Q_r\}$
A:

Then we design the following prompt for LLM self-judgment to get the verbal judgment O_j:

Prompt for LLM Self-Judgment

Given the following QA pair:
Q: $\{Q_r\}$
A: $\{O_r\}$
Does the answer above reflect the facts? Please respond with one of the following labels: "Yes" or "No".
Answer:

After generation, we filter out a small number of invalid responses (*e.g.*, '...'), which fall outside the scope of hallucination detection.

A.3 Automatic Labeling of LLMs' Responses

To obtain the factuality label L_r for LLM responses, we design a three-stage automated annotation pipeline consisting of text pattern matching, semantic similarity scoring, and annotation using GPT-4o-mini. Each stage hierarchically processes the remaining unlabeled samples.

First, for samples that cannot be annotated via text pattern matching, we compute the semantic entailment score between the LLM response and the golden answer using the nli-deberta-v3-base model[2] , which is specifically fine-tuned for natural language inference. Next, for samples with neutral NLI scores, we employ GPT-4o-mini as the annotator to assign factuality labels, using the prompt template shown below.

Prompt for GPT-4o-mini Annotation

Given the following Query–Response pair:
Query: $\{Q_r\}$
Response: $\{O_r\}$
A possible ground-truth answer is also provided:
Ground-Truth (possible): $\{G_t\}$
Your task is to determine whether the Response is accurate, based on:
1) The provided ground-truth (possible),
2) Your own knowledge.
Please choose one of the following labels as your final judgment:
"`True`", "`False`", "`Uncertain`".
Final Judgment:

For samples labeled as "Uncertain" by GPT-4o-mini, we conduct manual verification and selectively discard them due to the ambiguity and potential unreliability of their factuality labels. To assess annotation quality, we randomly sample 200 such uncertain instances for manual inspection, achieving an agreement rate of 93.50%, which indicates that GPT-4o-mini provides sufficiently reliable annotations for effective data filtering.

Finally, to further check the overall annotation quality, we randomly sample 800 instances from the automatically labeled dataset for manual verification. The agreement between automatic and human annotations reaches 96.125%, demonstrating the high quality of the automated labeling pipeline. The statistics of the final available dataset and label distribution are presented in Table 6.

B Details of Intrinsic Patterns Extraction

B.1 Hidden States (H_r & H_j)

Selection Strategy of K_{val}. To balance performance and efficiency while avoiding the increasing complexity of searching for the optimal layer as LLM size grows, we adopt a quantile-based strategy. Specifically, we partition the total number of LLM layers according to proportional quantiles $[1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8, 1]$, yielding eight candidate layers. During the implementation of SAPLMA, we perform a grid search over the hidden states of these candidate layers, and select the one achieving the highest Macro-F1 score on the validation set as $K_{\text{val},r}$ and $K_{\text{val},j}$ for the response and self-judgment settings, respectively.

B.2 Prediction Logits (P_r & P_j)

Additional Details of Logits Lens. Given a natural language input X, we first tokenize it into a sequence of tokens $T = [t_0, t_1, \ldots, t_{N-1}]$ using the tokenizer associated with the target LLM, where N denotes the number of tokens. The token sequence is then mapped to embedding vectors $E = [e_0, e_1, \ldots, e_{N-1}]$ via the embedding matrix. Subsequently, E is processed through a stack of Transformer blocks to produce hidden states $H = [h_n^k]$, where h_n^k represents the hidden state of the n-th token at the k-th layer. The hidden state of the final token at the last layer, h_{N-1}^K, is projected through the unembedding matrix to obtain the probability distribution for next-token prediction.[3]

For Logits Lens, let N_{Q_r} and N_{O_r} denote the token lengths of the query Q_r and response O_r,

[2] https://huggingface.co/cross-encoder/nli-deberta-v3-base

[3] For brevity, we omit standard operations such as Layer Normalization and Softmax.

Table 6: Dataset statistics across benchmarks and large language models used in the experiments

LLM	TriviaQA			MMLU			NQ_Open			HaluEval		
	Total	Real	Hallu.	Total	Real	Hallu.	Total	Real	Hallu.	Total	Real	Hallu.
Llama-3.1-8B-Instruct	9,668	6,981	2,687	10,555	6,863	3,692	3,255	2,183	1,072	8,976	3,784	5,192
Llama-3.1-70B-Instruct	9,848	8,485	1,363	9,112	7,518	1,594	3,400	2,638	762	9,438	5,225	4,213
Qwen2.5-32B-Instruct	9,702	7,161	2,541	10,574	8,643	1,931	3,364	2,147	1,217	9,296	4,345	4,951
Mistral-7B-Instruct-v0.3	9,752	6,859	2,893	10,750	6,612	4,138	3,316	2,072	1,244	9,203	3,970	5,233

respectively. To compute the probabilities of tokens in O_r, we first extract the hidden states corresponding to the token span $T[N_{Q_r} - 1 : N_{Q_r} + N_{O_r} - 1]$ across all layers, resulting in a tensor of shape $(N_{O_r}, \text{layer\_num}, \text{hidden\_dim})$. We then project these hidden states into the LLM's output space using the unembedding matrix, yielding a probability tensor of shape $(N_{O_r}, \text{layer\_num}, \text{vocab\_size})$.Finally, we obtain the layer-wise probabilities for each token in O_r by indexing this tensor with the corresponding token IDs, resulting in a matrix of shape $(N_{O_r}, \text{layer\_num})$.[4]

To obtain a sequence-level Logits prediction representation, we employ a single multi-head Transformer layer followed by mean pooling to aggregate token-level representations in P_r. Similar design choices have been explored in prior works such as those from Wang et al. (2023) and Shi et al. (2024). For P_j, we extract the Logits Lens representation of the first token in O_j (typically "Yes" or "No", along with their aggregated synonyms), and further enhance the contrast using Eq. (1). The resulting representation is then used as a sequence-level feature, as it sufficiently captures the core decision semantics of the judgment.

The equivalents of *"Yes"* include:

```
['Yes', 'yes', 'YES', '_Yes', '_yes', '_YES', 'Y', 'y', '_Y', '_y', 'True', 'true', 'TRUE', '_True', '_true', '_TRUE', 'Correct', 'correct', 'CORRECT', '_Correct', '_correct', '_CORRECT']
```

The equivalents of *"No"* include:

```
['No', 'no', 'NO', '_No', '_no', '_NO', 'N', 'n', '_N', '_n', 'False', 'false', 'FALSE', '_False', '_false', '_FALSE', 'Incorrect', 'incorrect', 'INCORRECT', '_Incorrect', '_incorrect', '_INCORRECT']
```

Synonyms Group for "Yes"/"No". We adopt a verbalization strategy for LLM self-judgments to more accurately capture the model's decision uncertainty. Specifically, following common practices in prompt engineering, we treat tokens with similar semantics or prefixes as equivalents of "Yes" or "No". The synonym groups for "Yes" and "No" in Section 4.2 are constructed by intersecting the above token lists with the vocabulary of the target LLM.

B.3 Attention Scores (A_r & A_j)

Context Segmentation. Building upon the core idea of the original "Lookback" approach, we adapt the context segmentation strategy to align with the specific prompt templates utilized for the Response and Self-Judgment tasks in our work. For the response scenario (r), we partition the context into four distinct segments: system prompt, query Q_r, response trigger, and preceding tokens of O_r. The first three segments correspond directly to the three respective lines of the "Prompt for Response Generation" template detailed in Appendix A.2. Similarly, for the self-judgment scenario (j), we partition the context into six distinct segments: Framing, Query, Response, Eval_Query, Format, and Trigger. These six segments correspond respectively to the six clauses of the "Prompt for LLM Self-Judgment" template, also detailed in Appendix A.2.

Lookback Ratio Calculation. We follow the core implementation by Chuang et al. (2024) and adapt it to our task setting. Specifically, we first partition all tokens in a sequence into two categories: *anchor tokens* and *context tokens*. We then extract the token-level attention score matrix from anchor tokens to context tokens, with shape $(\text{layer\_num} \times \text{head\_num}, N_{\text{ach}}, N_{\text{ctx}})$, where head_num denotes the number of attention heads per layer in the LLM, and N_{ach} and N_{ctx} represent the number of anchor tokens and context tokens, respectively. Next, according to the aforementioned context segmentation scheme, we divide the context tokens into N_{seg} segments. For each anchor token, we perform intra-segment average pooling over the attention scores

[4] vocab_size denotes the vocabulary size of the LLM, and *layer_num* corresponds to K defined above.

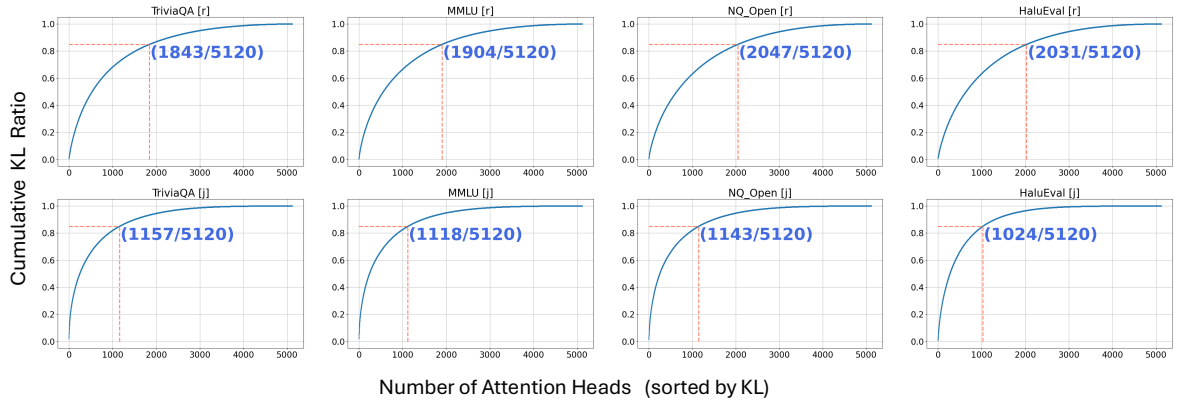

Figure 5: Normalized cumulative distribution of KL divergence across all attention heads in Llama-3.1-70B. The pronounced long-tail distribution reveals that a small subset of heads captures the core factual discriminative capacity, while the majority provides only weak signals. The orange dashed line indicates a cumulative probability of 0.85.

assigned to the context tokens within each segment, yielding *ach2seg* attention distributions with shape $(\text{layer\_num} \times \text{head\_num}, N_{\text{ach}}, N_{\text{seg}})$. We then normalize the *ach2seg* attention map along the N_{seg} dimension to obtain the token-level Lookback Ratio. Finally, we apply average pooling over all anchor tokens to derive the sequence-level Lookback Ratio, with shape $(\text{layer\_num} \times \text{head\_num}, N_{\text{seg}})$. For the response scenario (r), the anchor is O_r, the context is concat(Q_r, O_r), and $N_{\text{seg}} = 4$. For the self-judgment scenario (j), the anchor is O_j ("Yes/No"), the context is Q_j, and $N_{\text{seg}} = 6$.

Top-P Informative Heads Selection. As described above, the Lookback Ratio yields a high-dimensional vector of shape $(\text{layer\_num} \times \text{head\_num}, N_{\text{seg}})$ for each sequence. As LLMs scale, this dimensionality increases substantially (e.g., from 1024 heads in Llama-3.1-8B to 5120 in Llama-3.1-70B), raising the question of potential redundancy in these representations.

Specifically, we compute the KL divergence between the Lookback Ratio distributions of positive and negative training samples to quantify each head's factual discriminative power. We then rank the heads by descending KL divergence and plot the normalized cumulative distribution (as shown in Figure 5). The curve exhibits a sharp initial increase before flattening, indicating a pronounced long-tail distribution. This reveals that a small subset of heads captures the core factual discriminative capacity, while the majority provides only weak signals. Figure 5 only shows results on Llama-3.1-70B for brevity, and consistent patterns hold across three other LLMs.

Such sparsity and redundancy limit effective training and introduce unnecessary computational overhead. Consequently, we apply a top-P selection strategy ($P = 0.85$, orange dashed line in Figure 5) to retain only heads with strong factual discriminability. Empirically, this approach preserves the hallucination detector's performance while effectively mitigating the long-tail effect and reducing the classifier's input dimensionality.

C Huber Loss Function

The Huber loss (Huber, 1964) is a robust loss function that combines the advantages of Mean Squared Error (MSE) and Mean Absolute Error (MAE). Given a prediction x and target y, the Huber loss is defined as:

$$\mathcal{L}_\delta(x, y) = \begin{cases} \frac{1}{2}(x-y)^2, & \text{if } |x-y| \le \delta, \\ \delta\left(|x-y| - \frac{1}{2}\delta\right), & \text{otherwise,} \end{cases} \tag{8}$$

where δ is a threshold hyperparameter that controls the transition between quadratic and linear regimes. In our framework, we set the threshold hyperparameter to $\delta = 0.5$.

Compared to MSE, which applies a quadratic penalty to all errors and is sensitive to outliers, the Huber loss behaves quadratically for small errors and linearly for large errors. This property makes it more robust to noisy or misaligned signals, while still maintaining smooth optimization near the optimum.

D Large Language Models

We use four commonly used open-source LLMs to cover different model families and scales. For

Table 7: Efficiency profiling of three versions of LaaB. Parameters are reported in thousands (**K**). Inference latency is measured in **ms/instance**.

Version	Params (R/E) ($\times 10^3$)	Train Speed (s / epoch)	Inf. Latency (ms / inst.)
Hidden Based	1,090 / 1,090	1.06	0.0215
Logits Based	155 / 25	1.63	0.0347
Attns Based	575 / 520	0.40	0.0232

Llama 3 (Grattafiori et al., 2024), we use Llama-3.1-8B-Instruct[5] and Llama-3.1-70B-Instruct[6]. For Qwen-2.5 (Yang et al., 2025), we use Qwen-2.5-32B-Instruct[7]. For Mistral (Jiang et al., 2023), we use Mistral-7B-Instruct-v0.3[8]. The three LLM families are developed and released by independent organizations in different countries, and all of them are popular in the open-source community, which enhances their representativeness.

E Efficiency Analysis

We conducted experiments on a server equipped with a single NVIDIA A800 GPU with a batch size of 128 and a learning rate of 1e-4 for the efficiency test. As summarized in Table 7, the *Attns Based* variant demonstrates the highest training speed, requiring only 0.40 s/epoch. In terms of inference, the *Hidden Based* variant achieves the lowest latency at 0.0215 ms/instance. Notably, although the *Logits Based* variant has the fewest trainable parameters, it incurs higher temporal costs in both training (1.63 s/epoch) and testing phases compared to the other variants. This indicates that our proposed LaaB framework would not introduce a significant increase in inference cost, making it a practical option for training hallucination detectors.

F Introduction of Consistency-based Baselines

Here, we supplement the introduction of two hallucination detection baseline methods based on multi-sampling consistency:

- **SelfCheckGPT (Manakul et al., 2023):** A sampling-based hallucination detector based on the assumption that the sampled responses will be consistent if an LLM has clear "knowledge" to respond to a query. If the LLM is hallucinating, the responses obtained from multiple samplings would be inconsistent. SelfCheckGPT provides five implementations, including BERTScore, MGQA, Unigram, NLI, and GPT-Prompt. In this work, we adopt SelfCheckGPT-NLI to balance detection performance and efficiency.
- **Eigen-Score (Chen et al., 2024):** An uncertainty-based method to detect hallucinations by leveraging the semantic consistency of generated outputs in the dense embedding space. Specifically, it constructs a covariance matrix from the hidden states of multi-sampled responses and calculates its logarithmic determinant (equivalent to the sum of the logarithms of all eigenvalues) to measure the differential entropy of the representations. Therefore, a higher Eigen-Score indicates greater semantic divergence and a higher likelihood of hallucination. Following the original setup, we extract the hidden state of the final token from a middle layer for each sequence and report results without feature clipping in our experimental tables.

G Evaluation on More Baseline Methods

To further demonstrate the broad adaptability of the LaaB framework across diverse hallucination detection approaches, we augment our experiments with two additional baselines: LapEigvals and TSV. Specifically, we compare the detection performance of the base detectors against their LaaB-enhanced counterparts. A brief introduction to these baselines is provided below:

- **LapEigvals (Binkowski et al., 2025):** An attention-based method that leverages the spectral properties of LLM attention maps. Specifically, it interprets the attention maps generated during the autoregressive inference process as weighted adjacency matrices of directed graphs. For each attention head across all transformer layers, the method computes the corresponding graph Laplacian matrix and extracts its top-k eigenvalues. These eigenvalues serve to quantify disruptions or bottlenecks in the model's internal information flow, which are hypothesized to correlate with the occurrence of hallucinations. Ultimately, the extracted spectral

[5] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[6] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

[7] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Table 8: Performance comparison of LaaB with additional baseline methods (LapEigvals and TSV) in hallucination detection. The **blue-shaded** rows indicate the LaaB-enhanced versions. **Bolded** numbers denote that the use of LaaB outperforms its corresponding base version.

LLM	Method	TriviaQA		MMLU		NQ_Open		HaluEval		Average	
		macF1	Acc	macF1	Acc	macF1	Acc	macF1	Acc	macF1	Acc
Llama-3.1-8B-Instruct	LapEigvals	74.92	78.32	68.77	69.75	69.56	72.71	75.81	76.04	72.27	74.21
	+*LaaB*	**76.44**	**80.69**	**69.71**	**72.07**	**70.48**	**75.15**	**76.45**	**76.88**	**73.27**	**76.20**
	TSV	75.61	79.71	61.49	65.97	71.05	73.02	74.48	75.15	70.66	73.46
	+*LaaB*	**77.17**	**81.47**	**62.59**	**66.92**	**71.49**	**74.84**	**74.82**	**75.32**	**71.52**	**74.64**
Llama-3.1-70B-Instruct	LapEigvals	72.34	83.08	68.59	76.16	66.03	71.30	76.17	76.37	70.78	76.73
	+*LaaB*	**73.01**	**86.37**	**71.08**	**81.97**	**68.44**	**77.01**	**76.38**	**76.48**	**72.23**	**80.46**
	TSV	71.85	85.41	71.12	81.48	66.95	74.52	77.85	77.96	71.94	79.84
	+*LaaB*	**72.75**	**86.47**	71.10	**81.75**	**68.17**	**77.01**	**78.09**	**78.17**	**72.53**	**80.85**
Qwen-2.5-32B-Instruct	LapEigvals	74.59	78.55	64.15	72.29	76.31	77.55	76.40	76.49	72.86	76.22
	+*LaaB*	**76.54**	**81.53**	**65.86**	**77.15**	**78.82**	**80.06**	**77.01**	**77.08**	**74.56**	**78.96**
	TSV	76.21	80.86	63.27	69.18	76.47	77.55	77.20	77.29	73.29	76.22
	+*LaaB*	**78.12**	**83.18**	**64.21**	**71.19**	**78.30**	**79.32**	**77.82**	**77.94**	**74.61**	**77.91**
Mistral-7B-Instruct-v0.3	LapEigvals	74.18	76.66	68.27	68.99	73.57	75.11	73.41	73.75	72.36	73.63
	+*LaaB*	**74.91**	**78.05**	**68.31**	68.99	**74.59**	**76.16**	**74.08**	**74.67**	**72.97**	**74.47**
	TSV	76.69	79.53	68.16	69.96	75.90	76.76	76.20	76.46	74.24	75.68
	+*LaaB*	76.60	**79.58**	**69.14**	**70.24**	**76.50**	**77.66**	74.44	75.54	74.17	**75.76**

features are concatenated, projected to a lower-dimensional space using Principal Component Analysis (PCA), and fed into a supervised probe to predict the final hallucination label.

- **TSV (Park et al., 2025):** A hidden-based intervention method that reshapes the latent space of LLMs for hallucination detection. Pre-trained embeddings are optimized for linguistic coherence rather than factual accuracy, causing truthful and hallucinated representations to overlap. To mitigate this, TSV injects a single trainable steering vector into the residual stream of an intermediate transformer layer, scaled by a fixed strength hyperparameter. This lightweight intervention propagates through subsequent layers via inherent non-linear transformations, avoiding full model fine-tuning. As a result, the final-layer representations—often characterized by a von Mises-Fisher distribution—are reorganized into more compact and separable clusters, improving the linear separability between truthful and hallucinated outputs without degrading core language capabilities.

We follow the core designs of the two baseline methods and make light adaptations to align them with the LaaB framework. The detailed configurations are as follows:

LapEigvals. We extract the top-10 Laplacian eigenvalues for each attention head and reduce the concatenated spectral features to 512 dimensions via PCA. Both the response detector D_r and the self-judgment detector D_j use an MLP probe with hidden sizes $[128, 32]$.

TSV. The steering vector is injected into the Transformer residual stream with a strength of 5 and an EMA decay rate of 0.9. The intervention layer is chosen by model depth: layer 8 for Llama-3.1-8B-Instruct and layer 16 for Qwen2.5-32B-Instruct (approximately one-quarter depth), and layer 40 for Llama-3.1-70B-Instruct and layer 16 for Mistral-7B-Instruct-v0.3 (approximately halfway). We randomly sample 2,000 training instances to learn separate steering vectors for the response and self-judgment scenarios. The LaaB constraint is then applied to the final-layer hidden representations after steering. Both D_r and D_j employ an MLP probe with hidden dimensions $[256, 128, 64]$.

The results in Table 8 show that the LaaB framework improves the performance of both LapEigvals and TSV in most cases. This further demonstrates that, by introducing logical consistency constraints from both the response and self-judgment perspectives, LaaB can be reliably integrated with diverse hallucination detectors, highlighting its effectiveness and compatibility.

H Algorithm of LaaB

The pseudocode for the training and inference procedures of the LaaB hallucination detection framework is shown in Algorithm 1.

Algorithm 1 Training & Inference of LaaB

Require: Training set $\mathcal{S}$, Validation set $\mathcal{S}_{val}$, Testing set $\mathcal{S}_{test}$, Learning rates $(\eta_r, \eta_j, \eta_{tune})$, Early stopping patience P.
Ensure: Response detector $D_r(\theta_r)$, Self-Judgment detector $D_j(\theta_j)$.

procedure TRAINING($\mathcal{S}, \mathcal{S}_{val}$)

STAGE 1: ASYNCHRONOUS MUTUAL LEARNING */* D_r and D_j learn interactively */*

$stop_r \leftarrow$ False, $stop_j \leftarrow$ False
while not ($stop_r$ **and** $stop_j$) **do**
for mini-batch $B \in \mathcal{S}$ **do**
1. Extract intrinsic features F_r, F_j for B
if not $stop_r$ **then**
2. $\mathcal{L}_{\text{CE},r}, \mathcal{L}_{\text{Logic},r} \leftarrow \text{MutualLoss}(D_r, D_j, F_r, F_j, S, role = r)$
3. $\alpha_r \leftarrow \text{AdaptiveWeight}(\mathcal{L}_{\text{CE},r}, \mathcal{L}_{\text{Logic},r}, \theta_r)$
4. $\mathcal{L}_r \leftarrow \mathcal{L}_{\text{CE},r} + \alpha_r \mathcal{L}_{\text{Logic},r}$
5. $\theta_r \leftarrow \theta_r - \eta_r \nabla_{\theta_r} \mathcal{L}_r$
end if
if not $stop_j$ **then**
6. $\mathcal{L}_{\text{CE},j}, \mathcal{L}_{\text{Logic},j} \leftarrow \text{MutualLoss}(D_r, D_j, F_r, F_j, S, role = j)$
7. $\alpha_j \leftarrow \text{AdaptiveWeight}(\mathcal{L}_{\text{CE},j}, \mathcal{L}_{\text{Logic},j}, \theta_j)$
8. $\mathcal{L}_j \leftarrow \mathcal{L}_{\text{CE},j} + \alpha_j \mathcal{L}_{\text{Logic},j}$
9. $\theta_j \leftarrow \theta_j - \eta_j \nabla_{\theta_j} \mathcal{L}_j$
end if
end for
// Early Stopping & State Reversion
Evaluate BCE loss on $\mathcal{S}_{val}$
If D_r (D_j) does not improve for P epochs, freeze θ_r (θ_j) and set $stop_r$ ($stop_j$) $\leftarrow$ True
If one converges, revert to its best state and continue training the other
end while

STAGE 2: JOINT FINE-TUNING */* Synchronized optimization */*

Restore D_r and D_j to their best states from Stage 1 and unfreeze
$wait \leftarrow 0$, $stop \leftarrow$ False
while not $stop$ **do**
for mini-batch $B \in \mathcal{S}$ **do**
1. $\mathcal{L}_{\text{CE},r}, \mathcal{L}_{\text{CE},j}, \mathcal{L}_{\text{Logic}} = \text{JointLoss}(D_r, D_j, F_r, F_j, S)$
2. $\alpha \leftarrow \frac{1}{2} \sum_{k \in \{r,j\}} \text{AdaptiveWeight}(\mathcal{L}_{\text{CE},k}, \mathcal{L}_{\text{Logic}}, \theta_k)$
3. $\mathcal{L}_{\text{Joint}} \leftarrow \mathcal{L}_{\text{CE},r} + \mathcal{L}_{\text{CE},j} + \alpha \mathcal{L}_{\text{Logic}}$
4. $\theta_r \leftarrow \theta_r - \eta_{tune} \nabla_{\theta_r} \mathcal{L}_{\text{Joint}}$, $\theta_j \leftarrow \theta_j - \eta_{tune} \nabla_{\theta_j} \mathcal{L}_{\text{Joint}}$
end for
Evaluate $\mathcal{L}_{\text{CE},r}$ on $\mathcal{S}_{val}$: update best θ_r^*, θ_j^* if improved, else $wait \leftarrow wait + 1$
If $wait \geq P$ **then break** */* Early stopping */*
end while
return Optimal parameters θ_r^*, θ_j^*
end procedure

---

procedure INFERENCE($\{Q_r, O_r\} \in S_{test}$)
DEPLOYMENT ▷ Only D_r is deployed to save costs
1. Extract intrinsic feature $F_r \in \{H_r, P_r, A_r\}$ during generating O_r
2. Predict hallucination probability $S_r = D_r^*(F_r)$
return S_r
end procedure