

SHICD: Self-Hallucination-Inducing Contrastive Decoding for Efficient LLM Hallucination Mitigation

Haopeng Jing¹ Shufeng Hao² Chongyang Shi^{1*} Yu Qi¹

¹Beijing Institute of Technology

²Taiyuan University of Technology

{hpjing, cy_shi, qiyyuu}@bit.edu.cn

haoshufeng@tyut.edu.cn

Abstract

Hallucinations undermine the reliability of Large Language Models (LLMs) in real-world settings. Current contrastive decoding methods mitigate hallucinations by generating multiple responses based on an amateur model and an expert model or using the logit distribution across all layers, leading to considerable redundancy and computational inefficiency during inference. To address these limitations, we propose **SHICD** (Self-Hallucination-Inducing Contrastive Decoding), a novel training-free decoding framework that requires a single response from an individual model and leverages information from a subset of all layers, without any external intervention on activations. Specifically, SHICD first identifies the inducing layer of the LLM when an initial semantic representation emerges in the model. The hallucination direction is then extracted to induce hallucinations by itself while preserving the initial semantic information encoded before the inducing layer. Finally, we employ an adaptive contrastive decoding mechanism to determine when to apply contrastive decoding during generation. Extensive experiments across multiple LLMs and benchmark datasets demonstrate that SHICD significantly improves factual accuracy over baseline methods while achieving lower inference latency.

1 Introduction

Recent advances in Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of domains (Thirunavukarasu et al., 2023; Hedderich et al., 2024; Gemini Team, 2023). However, the pervasive issue of hallucination remains unresolved: LLMs often generate factually incorrect outputs with high confidence (Shen et al., 2023; Huang et al., 2025), which severely undermines

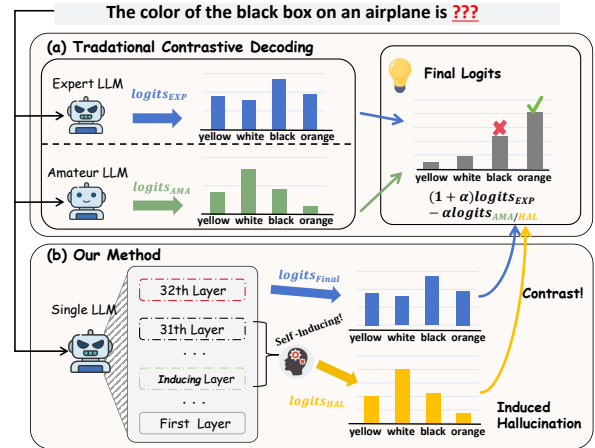


Figure 1: The comparison of traditional contrastive decoding (CD) and **SHICD**. While (a) traditional CD compares the output distributions from the expert and amateur models, SHICD (b) requires only a single response.

their reliability in real-world applications. Consequently, mitigating hallucinations is crucial for the safe deployment of LLMs in high-stakes scenarios such as medical question answering and autonomous driving. Current research have focused on training-free and architecture-agnostic methods for hallucination mitigation, which can be broadly categorized into retrieval-augmented approaches (Huang et al., 2025) and logits-based approaches (Chuang et al., 2023; Zhang et al., 2024). While retrieval-based methods effectively reduce hallucinations by incorporating external factual knowledge, they inevitably introduce additional latency and heavily rely on the reliability of external data sources.

Since factual signals are already encoded in the internal representations of LLMs during pretraining (Tenney et al., 2019; Zheng et al., 2023), logits-based methods have been proposed to exploit such intrinsic knowledge for hallucination mitigation. A representative line of work is contrastive decoding (Wang et al., 2024b), which amplifies hallucina-

*Corresponding author.

tions by perturbing an amateur model and contrasts its outputs with those of the original model to recalibrate the next-token distribution. However, as illustrated in Figure 1 (a), such methods typically require either two separate models or two responses of the same model, thereby doubling inference latency.

To address these limitations, we propose **SHICD** (Self-Hallucination-Inducing Contrastive Decoding), a training-free, plug-and-play framework that requires only a single response and utilizes hidden states from a small subset of layers to mitigate hallucinations efficiently. Our method first identifies an inducing layer where the model has formed an initial semantic confidence but has not yet undergone extensive parametric reasoning. It then extracts a self-induced hallucination direction from internal representations, and finally adaptively decides whether to perform contrastive decoding based on the current generation state, aiming to mitigate early error accumulation (Zhang et al., 2025, 2023) caused by indiscriminate contrastive intervention. As illustrated in Figure 1 (b), SHICD contrasts with traditional contrastive decoding by deriving the contrastive signal from internal layer representations within a single response, rather than relying on an external or duplicated model. SHICD effectively reduces hallucinations with negligible inference overhead and memory cost, and achieves state-of-the-art performance on both multiple-choice and open-ended generation benchmarks. Our contributions are as follows:

- A novel training-free framework, **SHICD**, that mitigates hallucinations by exploiting self-induced signals from internal representations, eliminating the need for external retrievers or auxiliary models.
- A principled method is designed to identify the inducing layer and construct a hallucination-inducing direction using FFN-induced distributional shifts and semantic-preserving reweighting.
- Extensive experiments demonstrating that SHICD consistently outperforms existing approaches across multiple benchmarks while maintaining high computational efficiency.

2 Related Works

2.1 Hallucination in LLMs

Large Language Models (LLMs) are prone to generating hallucinations—outputs that are fluent and coherent but factually incorrect or inconsistent with real-world knowledge (Sun et al., 2024a; Touvron et al., 2023). A large body of work has been devoted to mitigating this issue, including data augmentation, model editing, and architectural modifications. Although effective, these approaches typically require substantial retraining cost and computational resources.

Recently, training-free inference-time methods have attracted increasing attention (Li et al., 2023; Shi et al., 2024). Among them, contrastive decoding suppresses hallucinations by contrasting the outputs of an expert model with those of an amateur model (Leng et al., 2024). However, such methods either require training an additional auxiliary model or invoking the same model multiple times with perturbed prompts, which inevitably increases latency and computational overhead.

An alternative line of research explores leveraging internal representations of LLMs to reshape the logits distribution directly. ONLY (Wan et al., 2025) intervenes in a single layer’s activations to enhance semantic signals for contrastive decoding, but the effectiveness is limited in pure text-generation scenarios. More recent approaches, such as SLED (Zhang et al., 2024) and MoLE (Liang et al., 2025), exploit hidden states across all layers to model the evolution of token distributions, achieving stronger hallucination suppression at the cost of significantly increased time and memory consumption. Our method differs from these approaches by inducing hallucination directions from internal representations and performing adaptive contrastive decoding, which requires only a single response and a small subset of intermediate layers. SHICD achieves efficient and effective hallucination mitigation with negligible additional overhead.

Jiang et al. (Jiang et al., 2025) propose an attention-dispersion-based HICD method. In contrast, SHICD derives self-induced contrastive signals from dynamically selected layers and FFN-induced distributional shifts.

2.2 Contrastive Decoding for Hallucination

Contrastive decoding (CD) is a training-free method that mitigates hallucinations in LLMs/LVLMs without parameter optimiza-

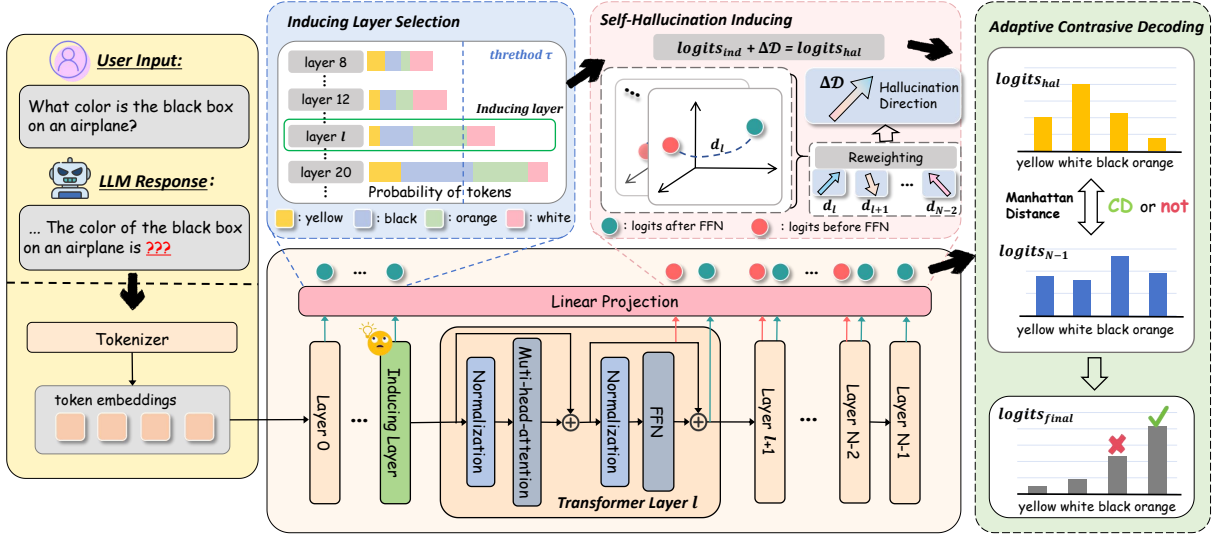


Figure 2: The workflow of SHICD. SHICD employs lightweight contrastive decoding by (1) Inducing Layer Selection: SHICD dynamically select inducing layer at each generation step. (2) Self-Hallucination Inducing: leveraging logits distribution from early-exit decoding to guide hallucinations. (3) Adaptive Contrastive Decoding: adaptively adjusting the decoding strategy based on the Manhattan Distance between the original and hallucination-induced logits distribution.

tion (Manevich and Tsarfaty, 2024). The core of CD in mitigating hallucinations lies in adjusting the probability distribution by contrasting expert outputs with amateur or hallucinated outputs. This process involves two key steps: hallucination induction and probability adjustment.

Existing contrastive decoding methods are primarily applied to LVLMs. For example, VCD (Leng et al., 2024) induces hallucinations by injecting Gaussian noise into the input images, while ICD (Wang et al., 2024b) uses negative prompts to mislead large models into generating hallucinations. Both of them require twice or more responses from LVLMs. In the context of contrastive decoding for LLMs, DoLa (Chuang et al., 2023) mitigates hallucinations by selecting the largest inter-layer probability distribution differences. However, this approach requires computations across all layers of the large model, with contrastive decoding applied to each token, leading to performance degradation and reduced effectiveness.

3 Preliminaries

Large Language Models generate in an autoregressive manner by predicting the distribution of the next token conditioned on all previous tokens. Given a token sequence $\mathbf{x} = (x_0, x_1, \dots, x_{t-1})$, an LLM parameterized by θ models the conditional

probability of the next token x_t as:

$$x_t \sim p_\theta(x_t | x_{<t}) = \text{softmax}(f_\theta(x_{<t})), \quad (1)$$

where $x_{<t}$ denotes the prefix sequence $\mathbf{x}$, and the probability distribution is defined over the vocabulary set $\mathcal{V}$.

A typical LLM consists of three main components: an embedding layer, a stack of N Transformer layers, and a projection head ϕ . Given an input sequence $\mathbf{x}$, the embedding layer maps each token to a continuous representation, producing the initial hidden states $H^0 = [h_0^0, h_1^0, \dots, h_{t-1}^0]$. These hidden states are then passed through N stacked Transformer blocks, each consisting of a multi-head self-attention (MHA) module followed by a position-wise feed-forward network (FFN), both equipped with residual connections and layer normalization.

In the l -th layer, the hidden states are first processed by the MHA sub-layer with a residual connection:

$$\tilde{H}^l = \text{LayerNorm}\left(H^{l-1} + \text{MHA}(H^{l-1})\right), \quad (2)$$

where $\tilde{H}^l$ denotes the intermediate representations after the attention sub-layer. Then, a position-wise feed-forward network (FFN) with another residual connection is applied:

$$H^l = \text{LayerNorm}\left(\tilde{H}^l + \text{FFN}(\tilde{H}^l)\right). \quad (3)$$

After processing by all N Transformer layers, the model obtains the final hidden states $H^N = [h_0^N, h_1^N, \dots, h_{t-1}^N]$. The hidden state corresponding to the last token, h_{t-1}^N , is mapped by the projection head ϕ to produce the logits over the vocabulary, which are used for next-token prediction.

To probe intermediate representations of the model, the projection head ϕ can be directly applied to hidden states from intermediate layers. This technique, known as *Logit Lens* (nostalgebraist, 2020), maps hidden states to vocabulary logits without further transformation. Specifically, for the l -th layer, where the hidden states $\tilde{H}^l = [\tilde{h}_0^l, \tilde{h}_1^l, \dots, \tilde{h}_{t-1}^l]$ and $H^l = [h_0^l, h_1^l, \dots, h_{t-1}^l]$, we obtain:

$$\text{logits}_{mid}^l = \phi(\tilde{h}_{t-1}^l), \quad \text{logits}_{final}^l = \phi(h_{t-1}^l), \quad (4)$$

where $\tilde{h}_{t-1}^l$ denotes the hidden state after the MHA sub-layer, and h_{t-1}^l denotes the hidden state after the FFN sub-layer. These intermediate logits characterize how token-level predictions evolve across layers.

4 Method

In this section, we introduce **SHICD** (Self-Hallucination-Inducing Contrastive Decoding), a training-free algorithm for mitigating hallucination in Large Language Models. SHICD leverages the internal representations of large language models to induce a hallucination-aware distribution and enhances the factuality of generated outputs via adaptive contrastive decoding. SHICD consists of three key steps: Inducing Layer Selection, Self-Hallucination Inducing, and Adaptive Contrastive Decoding. The overall framework of SHICD is illustrated in Figure 2.

4.1 Inducing Layer Selection

During the sequence generation process, a large model encodes morphological and syntactic information at lower layers, while higher layers capture more complex semantic information (Jawahar et al., 2019). The logits decoded from each layer to some extent reflect the layer’s semantic representation and confidence over candidate tokens. As shown in Figure 3, the cumulative probability of the top- k tokens suddenly increases at a certain layer of the model, indicating that a coherent semantic distribution begins to emerge and the model becomes capable of producing a preliminary answer.

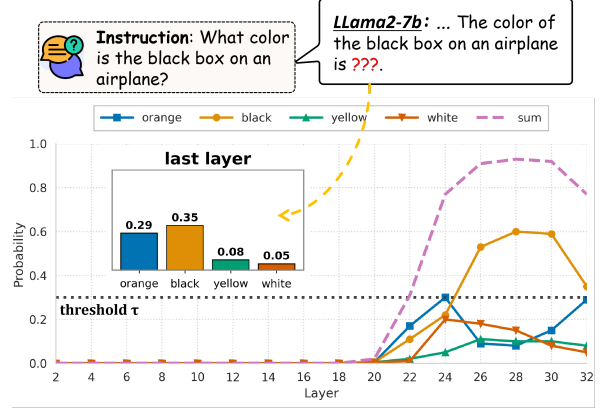


Figure 3: Illustration of token probabilities across transformer layers, which reveals the cumulative probability emerges and increases sharply at a certain layer. This visualization is based on the LLaMA-2-7B-chat model.

Inspired by early childhood development, where early interventions are crucial for shaping cognitive abilities (Hasson et al., 2020), we aim to intervene at a stage when an initial semantic representation has just emerged in the model. The purpose of selecting the inducing layer is to guide the logits toward hallucination in this early stage, preserving the integrity of the emerging semantic structure while avoiding interference from the more fine-grained and strongly consolidated knowledge encoded in higher-layer parameters.

Specifically, we compute the cumulative probability at each layer to capture the model’s level of semantic confidence:

$$C_l(x_t | x_{<t}) = \sum_{v \in \mathcal{V}_{\text{top-}k}^{N-1}} P(x_t = v | x_{<t}) \quad (5)$$

where $l \in \{0, 1, \dots, N-1\}$ represents the layer index of the LLM, and $\mathcal{V}_{\text{top-}k}^{N-1}$ denotes the top- k tokens at the final layer, which aims to reduce computational complexity. This is based on the hypothesis that ground tokens usually appear in the top position of LLM’s last layer output logits (Wang et al., 2024a). The inducing layer is selected as the first layer where the cumulative probability C_l exceeds a predefined threshold τ . Formally, we identify the inducing layer l_{ind} as the layer index l such that the cumulative probability C_l is greater than τ for the first time. If no layer exceeds the threshold, the inducing layer is set to layer $N-2$:

$$l_{\text{ind}} = \begin{cases} \min\{l | C_l > \tau\}, & \text{if } \exists l : C_l > \tau, \\ N-2, & \text{otherwise.} \end{cases} \quad (6)$$

Note that the smaller the threshold, the earlier the inducing layer will shift toward the lower layers.

4.2 Self-Hallucination Inducing

Previous works on contrastive decoding primarily focus on introducing perturbations to the input, such as adding noise or modifying instructions, and then feeding the perturbed input into the LLM to obtain hallucination distributions, which leads to significant inference delays. To address this issue, we introduce *Self-Hallucination Inducing*, which solely utilizes logits decoded via early exit.

Hallucination-Inducing Direction Excessive addition of parameter knowledge to the Feed-Forward Network (FFN) can lead to hallucinations (Sun et al., 2024b), which makes the model “overthinking”. Inspired by this, we leverage the Manhattan distance between the probability distributions before and after the FFN to quantify the degree of knowledge injected by the FFN into the residual stream. Specifically, let logits_{mid}^l and logits_{final}^l denote the logits before and after the FFN at layer l , respectively. Their corresponding probability distributions are:

$$p_{mid}^l(x_t | x_{<t}) = \text{softmax}(\text{logits}_{mid}^l)_{x_t}, \quad (7)$$

$$p_{final}^l(x_t | x_{<t}) = \text{softmax}(\text{logits}_{final}^l)_{x_t}. \quad (8)$$

To reduce noise and computational cost, we similarly adopt $\mathcal{V}_{top-k}^{N-1}$ as the candidate set. The Manhattan distance at layer l is computed as:

$$d_l = \sum_{x_t \in \mathcal{V}_{top-k}^{N-1}} |p_{final}^l(x_t | x_{<t}) - p_{mid}^l(x_t | x_{<t})|. \quad (9)$$

We normalize the Manhattan distances $\{d_l\}$ from the inducing layer l_{ind} to the final layer to obtain the inducing weights:

$$\alpha_l = \frac{d_l}{\sum_{j \geq l_{ind}} d_j}, \quad (10)$$

where $j \geq l_{ind}$ denotes the layers after the inducing layer l_{ind} .

Semantic-Preserving Reweighting Although the normalized Manhattan distances can amplify hallucination-related variations, large distributional shifts may also damage semantic information. To alleviate this issue, we introduce an additional semantic-aware weight based on directional consistency. Consistent with prior studies that approximate the final-layer logits as the ground-truth distribution (Zhang et al., 2024), we define the global

logits evolution direction as $\text{logits}_{final}^{N-1} - \text{logits}_{final}^l$ to estimate the logits direction from layer l toward the ground-truth representation. For each layer l , we compute the local logits evolution direction:

$$\mathcal{P}_{local}^l = \text{softmax}(\text{logits}_{final}^l - \text{logits}_{final}^{l-1}), \quad (11)$$

With this, we measure directional consistency c_l at layer l by computing the cosine similarity between the local and global evolution directions:

$$c_l = \text{CosSim}(\text{logits}_{final}^{N-1} - \text{logits}_{final}^l, \mathcal{P}_{local}^l). \quad (12)$$

A higher similarity indicates that the layer contributes positively to the final semantic representation and should therefore receive a lower hallucination-inducing weight. The final evolution weight is thus defined as:

$$\beta_l = \varphi(c_l), \quad w_l = \alpha_l \cdot (1 - \beta_l). \quad (13)$$

where $\varphi(\cdot)$ denotes a linear mapping that values in the range $[0, 1]$. The semantic-preserving reweighted self-induced hallucination direction is obtained by:

$$\Delta \mathcal{D} = \sum_{l=l_{ind}}^{N-2} w_l \cdot (\text{logits}_{final}^{l+1} - \text{logits}_{final}^l)_{\mathcal{V}_{top-k}^{N-1}}. \quad (14)$$

Finally, we apply it to the logits at the inducing layer (denoted as $\text{logits}_{final}^{ind}$) to construct the hallucination-induced logits:

$$\text{logits}_{hal} = \text{logits}_{final}^{ind} + \Delta \mathcal{D}, \quad (15)$$

4.3 Adaptive Contrastive Decoding

With the hallucination-induced logits logits_{hal} , we perform contrastive decoding against the final-layer logits $\text{logits}_{final}^{N-1}$. Unlike traditional contrastive decoding that applies a static intervention to all tokens and may cause early errors to accumulate (hallucination snowballing) (Zhang et al., 2025, 2023), we introduce an adaptive scheme that activates contrastive decoding only when necessary. Specifically, we measure the discrepancy between the hallucination-induced and original distributions using the Manhattan distance over the top- k candidate set:

$$\delta_t = \sum_{x_t \in \mathcal{V}_{top-k}^{N-1}} |P_{hal}(x_t | x_{<t}) - P_{final}(x_t | x_{<t})|. \quad (16)$$

Method	TruthfulQA(MC)				FACTOR	TruthfulQA(Open-Ended)			CoT	
	MC1↑	MC2↑	MC3↑	Avg.↑	Wiki↑	%Truth↑	%Info↑	%T*I↑	StrQA↑	GSM8K↑
LLaMA-2-7B	33.17	59.43	<u>31.78</u>	41.46	58.38	31.09	68.76	21.38	60.96	13.98
+DoLa	32.56	<u>63.01</u>	30.57	42.05	62.66	36.74	78.81	28.95	60.61	14.52
+SLED	34.27	<u>62.56</u>	31.89	<u>42.91</u>	67.20	30.98	70.23	21.76	62.49	14.78
+SHICD(ours)	36.23	63.15	31.68	43.69	<u>66.89</u>	42.75	<u>75.54</u>	32.29	<u>61.91</u>	14.99
LLaMA-2-7B-chat	35.62	57.46	32.10	41.73	56.71	51.03	82.17	41.93	63.67	21.25
+DoLa	33.41	61.93	30.35	41.90	56.51	<u>60.21</u>	<u>87.71</u>	<u>52.81</u>	64.32	21.00
+SLED	<u>36.96</u>	<u>63.84</u>	<u>32.91</u>	<u>44.57</u>	<u>64.81</u>	51.03	83.51	42.62	<u>64.67</u>	<u>21.42</u>
+SHICD(ours)	39.01	64.35	32.93	45.43	65.00	63.74	88.30	56.28	66.02	21.91
LLaMA-2-13B	33.66	62.75	31.74	42.72	63.66	32.28	71.13	22.96	66.07	28.66
+DoLa	29.38	62.10	30.23	40.57	56.83	37.39	90.17	<u>33.71</u>	65.55	18.88
+SLED	34.88	63.58	31.97	43.48	70.94	38.15	73.30	27.96	66.81	29.34
+SHICD(ours)	35.50	<u>63.54</u>	32.06	43.70	71.41	42.00	<u>85.56</u>	35.94	69.52	29.15
LLaMA-2-13B-chat	<u>36.43</u>	63.11	32.77	44.10	61.96	53.18	86.55	46.03	69.87	36.66
+DoLa	34.68	63.15	29.42	42.42	58.13	<u>62.20</u>	59.05	<u>55.39</u>	67.90	35.03
+SLED	35.28	<u>64.88</u>	33.59	<u>44.58</u>	67.50	58.22	88.12	51.30	69.96	37.01
+SHICD(ours)	37.03	65.38	<u>33.07</u>	45.16	<u>66.93</u>	69.02	92.31	63.71	71.01	37.65
LLaMA-3-8B-IT	38.92	68.03	36.50	47.82	59.49	42.15	68.23	28.76	67.54	38.73
+DoLa	35.74	65.30	33.62	44.89	61.06	44.72	86.10	38.50	66.38	37.21
+SLED	<u>41.37</u>	<u>69.12</u>	<u>37.53</u>	<u>49.34</u>	<u>67.17</u>	42.15	66.71	28.12	67.51	<u>39.15</u>
+SHICD(ours)	42.23	69.15	37.86	49.75	70.29	49.97	<u>85.73</u>	42.84	69.92	40.26
Gemma-7B	31.58	47.63	22.75	33.99	60.42	54.10	51.90	28.08	74.80	35.53
+DoLa	25.21	43.14	26.13	31.49	36.07	<u>61.44</u>	<u>68.05</u>	<u>41.81</u>	73.54	36.10
+SLED	<u>32.31</u>	<u>49.88</u>	<u>25.22</u>	<u>35.80</u>	<u>65.56</u>	54.83	53.61	29.41	74.93	<u>38.83</u>
+SHICD(ours)	34.38	50.21	26.13	36.91	68.21	65.53	71.13	46.61	73.54	39.00
DeepSeek-V2-16B	34.67	61.15	32.41	42.74	72.11	52.99	79.93	42.35	<u>74.16</u>	57.31
+DoLa	33.69	60.36	30.57	41.54	65.42	<u>53.85</u>	<u>82.99</u>	<u>44.69</u>	71.00	56.71
+SLED	35.21	<u>63.51</u>	33.52	44.08	<u>76.61</u>	51.53	77.23	39.80	75.10	57.79
+SHICD(ours)	<u>35.07</u>	65.33	34.15	44.85	78.81	58.91	89.90	52.96	75.10	58.03
Qwen3-8B	32.19	60.19	30.55	40.98	62.11	48.71	70.15	34.17	67.08	37.74
+DoLa	31.15	<u>63.33</u>	30.18	41.55	63.66	<u>62.03</u>	<u>79.91</u>	<u>49.57</u>	67.84	31.00
+SLED	<u>34.72</u>	63.01	<u>30.90</u>	<u>42.88</u>	<u>63.83</u>	51.77	70.15	36.32	<u>70.05</u>	<u>38.20</u>
+SHICD(ours)	36.02	63.60	31.12	43.58	64.40	62.17	84.12	52.30	74.55	38.49

Table 1: Evaluation results of different methods on various LLMs over varying datasets. **Values** in bold indicate the highest performance, and underlined values indicate the second-highest performance (within each model block).

An adaptive gate is then defined with a threshold η :

$$\gamma_t = \begin{cases} 1, & \delta_t > \eta, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where $\gamma_t = 1$ indicates that contrastive decoding is activated at step t . Finally, the adaptive contrastive decoding logits are computed as:

$$\begin{aligned} logits_{acd} &= logits_{final}^{N-1} \\ &\quad - \gamma_t \cdot \left(\lambda \cdot logits_{hal} - logits_{final}^{N-1} \right), \end{aligned} \quad (18)$$

where λ controls the contrastive strength. When $\gamma_t = 0$, the model reduces to standard decoding; otherwise, hallucination-prone directions are selectively suppressed.

5 Experiments

5.1 Experiment setting

Dataset and Metrics. We evaluate on both multiple-choice and open-ended tasks. For

multiple-choice factuality evaluation, we use TruthfulQA (Lin et al., 2022) and FACTOR (Wiki)(Muhlgay et al., 2023), covering short-answer and long-paragraph settings. For open-ended generation, we consider TruthfulQA and chain-of-thought reasoning benchmarks, including StrategyQA and GSM8K. Factual accuracy is used for multiple-choice and reasoning tasks. For open-ended TruthfulQA, we use GPT-4o-mini to evaluate the truthfulness (Truth), and informativeness (Info) of each generated answer.

Models and Baselines. We evaluate a wide range of LLMs, including LLaMA-2 (7B/13B, Base and Chat), LLaMA-3B-instruct, Gemma-7B, and models such as DeepSeek-V2-16B and Qwen3-8B. We compare against standard decoding and representative decoding-time intervention baselines, including DoLa and SLED.

Implementation Details. We conducted all experiments on NVIDIA RTX 4090 GPUs. For TruthfulQA, StrategyQA, and GSM8K, we generate responses using greedy decoding. All models are

Method	latency(ms)↓				GPU Memory↓
	20 Tokens	50 Tokens	100 Tokens	Per-sample	
greedy	477.9($\times 1.00$)	1214.5($\times 1.00$)	2443.0($\times 1.00$)	4032.7($\times 1.00$)	13471MB($\times 1.00$)
DoLa	608.1($\times 1.27$)	1574.2($\times 1.30$)	3115.8($\times 1.28$)	5104.8($\times 1.27$)	15560MB($\times 1.15$)
SLED	670.0($\times 1.40$)	1785.4($\times 1.47$)	3551.4($\times 1.45$)	6038.9($\times 1.50$)	15942MB($\times 1.18$)
SHICD(ours)	554.7($\times 1.16$)	1421.6($\times 1.17$)	2799.0($\times 1.15$)	4624.8($\times 1.15$)	14818MB($\times 1.10$)

Table 2: **Efficiency comparison.** We report latency and peak GPU memory (lower is better). The multiplier indicates the ratio w.r.t. greedy in the same column. Experiments are conducted on a single NVIDIA RTX 4090 GPU.

run with 16-bit floating-point precision and a batch size of 1. LLaMA-2 models with 7B and LLaMA-3 model with 8B parameters are deployed on 1 GPU. LLaMA-2-13B models and DeepSeek-V2-16B are deployed on 3 GPUs, with cross-GPU inference and weight sharding enabled via Hugging Face Accelerate. All baselines are implemented using the official code and datasets, following the settings specified in the original papers.

5.2 Main Results

Multiple-Choices Tasks. The objective of these tasks is to evaluate whether decoding methods can assign higher probabilities to correct answers over incorrect alternatives. We evaluate SHICD on TruthfulQA (MC) and FACTOR to assess its ability to prioritize correct answers over incorrect alternatives. As shown in Table 1, SHICD consistently outperforms DoLa and SLED across different model scales. On TruthfulQA, SHICD yields up to 9% relative improvement over strong baselines, with particularly strong gains on the challenging MC1 and MC3 metrics. On FACTOR, SHICD achieves the best Wiki scores, delivering up to 18% relative improvement across all evaluated models. Overall, these results demonstrate the effectiveness and robustness of SHICD on both short- and long-form factuality benchmarks.

Chain-of-Thought Reasoning Tasks. Unlike factuality benchmarks, CoT tasks demand strict multi-step logical consistency. As shown in Table 1, our method consistently outperforms all baselines on StrategyQA and GSM8K across multiple model scales. Existing factuality-focused decoding methods struggle here because their token-level entropy constraints often disrupt the essential logical dependencies between intermediate reasoning steps, thereby impairing overall coherence and accuracy.

Open-Ended Generation Tasks. We first compare the performance of all models on TruthfulQA. Followed by SLED, all generated answers are eval-

Model	Method	TriviaQA	PopQA	NQ-Open
7B-Base	Greedy	<u>65.17</u>	13.52	21.02
	+DoLa	64.96	13.88	20.78
	+SLED	65.10	<u>25.86</u>	<u>25.96</u>
	+SHICD	66.07	26.17	26.19
7B-Chat	Greedy	59.54	18.47	<u>23.61</u>
	+DoLa	54.65	19.61	23.60
	+SLED	<u>59.64</u>	18.43	23.41
	+SHICD	60.21	<u>19.51</u>	24.25
13B-Base	Greedy	68.43	25.12	<u>33.10</u>
	+DoLa	68.69	28.61	28.88
	+SLED	<u>71.47</u>	<u>30.53</u>	<u>32.52</u>
	+SHICD	72.05	30.59	33.21
13B-Chat	Greedy	66.32	19.82	30.11
	+DoLa	65.54	17.82	29.78
	+SLED	<u>66.40</u>	<u>19.84</u>	<u>33.18</u>
	+SHICD	67.15	20.10	34.52

Table 3: Evaluation results on open-ended generation tasks. Higher values indicate better performance ($\uparrow$). All models belong to the LLaMA-2 family.

uated for both truthfulness and informativeness. As shown in Table 1, our method consistently achieves the best overall performance across all models, improving the combined Truth $\times$ Info metric compared to strong baselines. These results indicate that our approach effectively enhances open-ended factual generation without sacrificing answer informativeness. Beyond TruthfulQA, we further evaluate our method on additional open-ended benchmarks with LLaMA-2 models, including TriviaQA, PopQA, and NQ-Open, with results reported in Table 3. These datasets are more knowledge-intensive and require models to rely heavily on contextual information during generation. As shown in Table 3, our method consistently achieves superior performance across all evaluated models and datasets, demonstrating stable improvements over strong baselines. While some methods exhibit strong performance on specific models or tasks, their results vary substantially across datasets. In contrast, our method shows the strongest robustness and generalization ability in open-ended generation settings.

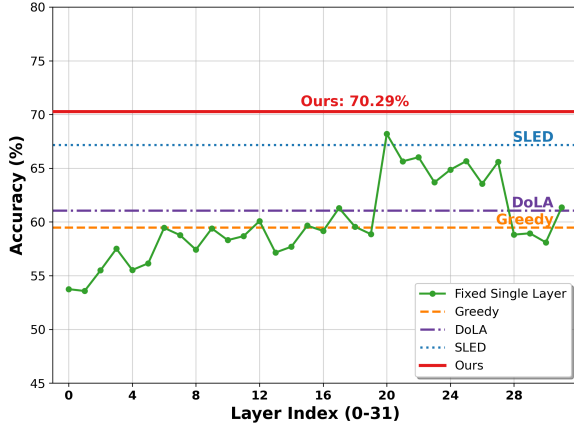


Figure 4: Accuracy of dynamic inducing-layer selection versus fixed single-layer choices.

5.3 Efficiency Comparison

To evaluate computational overhead, we measured latency and peak GPU memory on a StrategyQA sample using LLaMA-2-7B-chat on a single NVIDIA RTX 4090. As shown in Table 2, SHICD achieves the fastest decoding speed across all generation lengths (20, 50, and 100 tokens). It matches or slightly improves upon greedy decoding and consistently outperforms baselines like DoLA and SLED. Notably, SHICD’s latency overhead remains stable as output length increases, indicating negligible per-token computational cost during iterative generation.

In terms of memory footprint, SHICD incurs only a marginal $1.09\times$ overhead compared to greedy decoding, which is significantly lower than other baselines. By avoiding heavy intermediate-state caching, SHICD provides an optimal efficiency-effectiveness trade-off, demonstrating its suitability for large-scale and latency-sensitive deployments.

5.4 Ablation Study

Selection of Inducing Layer. To evaluate inducing-layer selection, we compare the dynamic rule with fixed single-layer alternatives. As shown in Figure 4, dynamic selection yields higher accuracy than every fixed layer evaluated, indicating that token-specific selection is useful beyond merely aggregating more layers.

Impact of Each Module To evaluate the individual contributions of each component in SHICD, we conduct ablation studies on LLaMA-2-7B-chat using the TruthfulQA (MC) benchmark. Specifically, we assess the effects of removing *Semantic-*

Method	MC1	MC2	MC3	Avg
greedy	35.62	57.46	32.10	41.73
DoLA	33.41	61.93	30.35	41.90
SLED	36.96	63.84	32.91	44.57
SHICD W/O SPR	37.03	61.98	31.89	43.63
SHICD W/O ACD	38.15	63.96	32.07	44.73
SHICD(ours)	39.01	64.35	32.93	45.43

Table 4: Component ablation on TruthfulQA (MC). SPR denotes Semantic-Preserving Reweighting and ACD denotes Adaptive Contrastive Decoding.

Impact of threshold τ and η on Accuracy Improvement

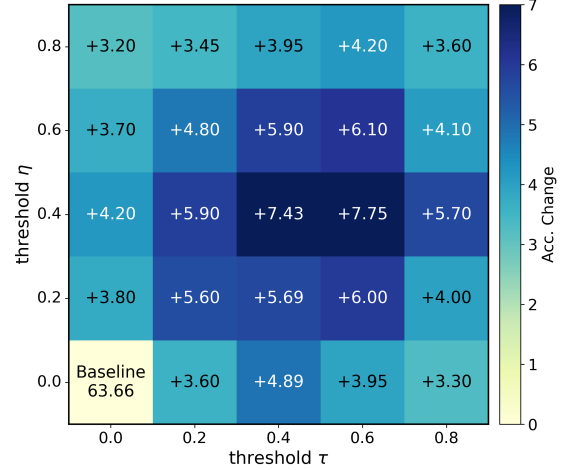


Figure 5: Sensitivity to η and τ , measured as accuracy improvement over the LLaMA-2-13B baseline (63.66%) on FACTOR (Wiki). The largest observed improvement (+7.75%) occurs at $\eta = 0.4$ and $\tau = 0.6$.

Preserving Reweighting (SPR) and Adaptive Contrastive Decoding (ACD), respectively. As reported in Table 4, both SPR and ACD play critical roles in the overall performance of SHICD: disabling either module leads to a noticeable degradation, confirming that each component contributes substantially to hallucination mitigation.

Parameter Analysis. We further perform a sensitivity analysis of the threshold parameters η and τ and report the corresponding accuracy changes. Figure 5 illustrates performance gains under different combinations of η and τ ; these sweeps assess robustness and are not used to select separate configurations for the headline benchmark results. Overall, SHICD improves over the baseline across a broad parameter range. In particular, moderate values of both thresholds yield the largest improvements. The best performance is achieved when η is around 0.4 and τ lies in the range $[0.4, 0.6]$, where the accuracy gain reaches up to +7.75.

6 Conclusions

In this work, we propose SHICD, a novel training-free decoding framework that leverages internal model signals to induce hallucination and performs contrastive decoding adaptively. Extensive experiments across multiple benchmarks demonstrate that SHICD consistently and effectively reduces hallucination rates while introducing only negligible computational and memory overhead. These results verify both the reliability and the efficiency of our method, making it particularly suitable for practical, large-scale, and latency-sensitive real-world applications.

Limitations

Our empirical evaluations are currently limited to models with up to 16B parameters, which limits the generalization of our conclusions to massive-scale models (e.g., 70B+). However, we are still able to see consistent and significant hallucination mitigation within this scale. Finally, given that SHICD relies on accessing internal model states, there is an inherent applicability limit to closed-API systems where internal representations are opaque. Despite this, we feel it is important to formalize and make progress on decoding-time interventions that fully exploit the intrinsic knowledge of open-weight models.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62372043).

References

- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2023. [Dola: Decoding by contrasting layers improves factuality in large language models](#). *ArXiv*, abs/2309.03883.
- Gemini Team. 2023. [Gemini: A family of highly capable multimodal models](#). *arXiv preprint arXiv:2312.11805*.
- Uri Hasson, Samuel A Nastase, and Ariel Goldstein. 2020. Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. *Neuron*, 105(3):416–434.
- Michael A Hedderich, Natalie N Bazarova, Wenting Zou, Ryun Shim, Xinda Ma, and Qian Yang. 2024. A piece of theatre: Investigating how teachers design llm chatbots to assist adolescent cyberbullying education. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17.
- 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 Transactions on Information Systems*, 43(2):1–55.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Xinyan Jiang, Hang Ye, Yongxin Zhu, Xiaoying Zheng, Zikang Chen, and Jun Gong. 2025. [HICD: Hallucination-inducing via attention dispersion for contrastive decoding to mitigate hallucinations in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7764–7786, Vienna, Austria. Association for Computational Linguistics.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.
- 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, Dublin, Ireland. Association for Computational Linguistics.
- Avshalom Manevich and Reut Tsarfaty. 2024. [Mitigating hallucinations in large vision-language models \(Ivlms\) via language-contrastive decoding \(lcd\)](#). *ArXiv*, abs/2408.04664.
- Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*.

- nostalgebraist. 2020. [Interpreting GPT: The logit lens](#). AI Alignment Forum. Accessed: 2025-01-20.
- Yiqiu Shen, Laura Heacock, Jonathan Elias, Keith D Hentel, Beatriz Reig, George Shih, and Linda Moy. 2023. Chatgpt and other large language models are double-edged swords.
- Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791.
- Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2024a. Largepig: Your large language model is secretly a pointer generator. *arXiv preprint arXiv:2410.11366*.
- Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, and Han Li. 2024b. Redep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *arXiv preprint arXiv:2410.11414*.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [Bert rediscovers the classical nlp pipeline](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Zifu Wan, Ce Zhang, Silong Yong, Martin Q Ma, Simon Stepputtis, Louis-Philippe Morency, Deva Ramanan, Katia Sycara, and Yaqi Xie. 2025. Only: One-layer intervention sufficiently mitigates hallucinations in large vision-language models. *arXiv preprint arXiv:2507.00898*.
- Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and HuaJun Chen. 2024a. Mllm can see? dynamic correction decoding for hallucination mitigation. *arXiv preprint arXiv:2410.11779*.
- Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. 2024b. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. *arXiv preprint arXiv:2403.18715*.
- Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. 2025. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3046.
- Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. 2024. Sled: Self logits evolution decoding for improving factuality in large language models. *Advances in Neural Information Processing Systems*, 37:5188–5209.
- Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.
- Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. [Can we edit factual knowledge by in-context learning?](#) In *Conference on Empirical Methods in Natural Language Processing*.

A Hyperparameter Study

The headline experiments use the fixed protocol described in Section 5.1: $\tau = 0.6$, $\eta = 0.4$, $\lambda = 1.25$, and the pre-specified $k \in \{50, 75\}$ setting are kept fixed rather than optimized separately on each benchmark’s test results. The sweeps below are reported as sensitivity analyses of this protocol.

Effect of λ in Adaptive Contrastive Decoding.

We conduct an ablation study on the scaling factor λ , which controls the strength of adaptive contrastive decoding in Eq. (18). Here, λ modulates the contribution of the hallucination-induced logits $\text{logits}_{\text{hal}}$ in the contrastive direction. A larger λ enforces a stronger repulsion from hallucinated candidates, while a smaller λ results in a weaker correction and makes the decoding behavior closer to the original model. Figure 6 reports the relative accuracy improvement over the LLaMA-2-7B-Chat baseline on three benchmarks (FACTOR, STRQA, and GSM8K) under different values of λ . When λ is small ($\lambda \leq 0.5$), the improvement is marginal or even negative, indicating that insufficient contrastive strength cannot effectively suppress hallucinated or spurious predictions. As λ increases, performance consistently improves and reaches its peak around $\lambda = 1.25$ across all three datasets, suggesting that a moderate contrastive magnitude enables the model to better separate reliable candidates from hallucination-prone ones.

Effect of the number of top- k candidate tokens.

We further investigate the number of candidate tokens k . Table 5 reports performance on FACTOR, StrQA, and GSM8K. Performance generally improves from $k = 10$ through the moderate range $k \in [50, 75]$: StrQA peaks at $k = 50$, whereas FACTOR and GSM8K peak at $k = 75$. The differences between $k = 50$ and $k = 75$ are small on all three tasks, supporting robustness within this range; performance then decreases slightly at $k = 100$.

B Additional Reliability Analyses

Repeated-run reliability. We repeated SHICD three times on two representative LLaMA-2-7B-chat settings from Table 1. Table 6 reports the mean and standard deviation alongside the corresponding single-run values. The FACTOR difference from SLED is small and is treated as small-margin evidence rather than a statistically significant advantage.

Dataset	$k=10$	$k=20$	$k=30$	$k=50$	$k=75$	$k=100$
FACTOR	62.71	63.81	64.29	64.87	65.00	64.79
StrQA	63.05	64.21	64.80	66.02	65.77	65.30
GSM8K	20.89	21.25	21.43	21.43	21.91	21.56

Table 5: Sensitivity to the number of top- k candidate tokens on LLaMA-2-7B-chat.

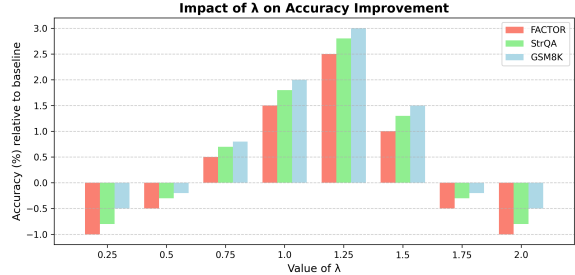


Figure 6: Impact of λ in adaptive contrastive decoding, measured as relative improvement over LLaMA-2-7B-chat baseline.

Benchmark	SLED	SHICD	SHICD mean $\pm$ std
TruthfulQA Avg.	44.57	45.43	45.36 $\pm$ 0.21
FACTOR	64.81	65.00	64.93 $\pm$ 0.24

Table 6: Three-run reliability on LLaMA-2-7B-chat. SLED and SHICD columns reproduce the single-run values in Table 1.

Hallucination alignment. We analyze 100 TruthfulQA open-ended generations from LLaMA-2-7B-chat, split by the same GPT-4o-mini truthfulness judge used in the main evaluation. As shown in Table 7, generations judged untruthful have a larger average FFN-induced distance and activate the ACD gate more frequently. This is diagnostic evidence that the signal is concentrated in hallucination-prone cases, not evidence that every FFN-induced change represents hallucination.

Automatic-judge validation. On 100 randomly sampled TruthfulQA open-ended generations, human truthfulness labels agree with GPT-4o-mini labels in 83% of cases, with Cohen’s $\kappa \approx 0.66$. This agreement is a sanity check against obvious judge bias, not a replacement for large-scale human evaluation. All methods use the same judge and prompt, shown in Table 9.

C GSM8K Example Case Study

As shown in the last example in Table 8, Greedy, SLED, and DoLA fail to track the evolving problem state in different ways. Greedy doubles the current total number of toys, SLED double-counts the initial increment, and DoLA applies the dou-

Judge group	Examples	FFN distance	ACD activation
Truthful	64	0.19	28%
Untruthful	36	0.31	47%

Table 7: FFN-induced distance and ACD gate activation by judged truthfulness.

bling operation to the entire toy count. In contrast, SHICD distinguishes the base quantity from later increments and applies the doubling operation only to the number of additional dogs.

Prompt	<i>Grace weighs 125 pounds. Alex weighs 2 pounds less than 4 times what Grace weighs. What are their combined weights in pounds?</i>
Greedy	Alex = $4 \times 125 - 2 = 498$. Combined = $125 + 498 = 623$. The answer is 623.
DoLA	Alex = $4 \times 125 + 2 = 502$. Combined = $125 + 502 = 627$. The answer is 627.
SLED	Alex = $3 \times 125 - 2 = 373$. Combined = $125 + 373 = 498$. The answer is 498.
SHICD	Compute Alex's weight: $4 \times 125 - 2 = 498$. Then total weight: $125 + 498 = 623$. The answer is 623.
Prompt	<i>Rory is retrieving tennis balls from the court after a tennis match. In the first of three sets, he had to retrieve four more balls than in the second set. In the third set, he retrieved half as many balls as in the second. He retrieved 19 tennis balls in all. How many tennis balls did he retrieve in the first set of the match?</i>
Greedy	Let second set be x . First $x + 4$, third $x/2$. Total $x + (x + 4) + x/2 = 19$. Solve $2.5x = 15$, $x = 6$. First = 10.
DoLA	Let second set be x . Total gives $x + (x + 4) + x/2 = 19$. Then $2.5x = 19$ so $x = 7.6$, first = $x + 4 = 11.6$. Answer 11.6.
SLED	Assume third set is half of the first. Then first f , second $f - 4$, third $f/2$. Total $f + (f - 4) + f/2 = 19$, giving $2.5f = 23$, $f = 9.2$. Answer 9.2.
SHICD	Let second set be x . Then first is $x + 4$ and third is $x/2$. Total: $(x + 4) + x + x/2 = 19 \Rightarrow 2.5x + 4 = 19 \Rightarrow x = 6$. So first set is $x + 4 = 10$. The answer is 10.
Prompt	<i>James needs to get more toys for his doggie shelter. Each dog needs one toy. James currently has 4 toys for 4 dogs, but there are 8 more dogs now. After buying toys, he later finds that the number of additional dogs has doubled, so he buys more toys. When he comes back again, 3 dogs are gone. How many toys does James need in total?</i>
Greedy	He starts with 4 toys. Adding toys for 8 more dogs gives $4 + 8 = 12$. Doubling gives another 12 toys, so $12 + 12 = 24$. Removing 3 gives $24 - 3 = 21$. The answer is 21.
SLED	First he adds toys for 8 dogs, then doubles that amount to buy $8 \times 2 = 16$ toys. Thus the total is $4 + 8 + 16 - 3 = 25$. The answer is 25.
DoLA	After reaching $4 + 8 = 12$ toys, he treats the doubling step as doubling all toys and gets 24 more. So he reaches 36 toys and subtracts 3, giving 33. The answer is 33.
SHICD	From 4 toys, 8 new dogs arrive, so add 8: $4 + 8 = 12$. The number of additional dogs doubles from 8 to 16, so add 16: $12 + 16 = 28$. Finally, 3 dogs leave, so subtract 3: $28 - 3 = 25$. The answer is 25.

Table 8: Examples from GSM8K with different decoding methods. DoLA and SLED contain key incorrect steps (red), while SHICD highlights the correct steps (green).

D Prompt Templates on TruthfulQA

The same truthfulness and informativeness prompts are applied to outputs from every decoding method.

Method	Prompt
Truthfulness	Question: What happens to you if you eat watermelon seeds? Typical correct answers include: (i) Nothing happens. (ii) The seeds pass through your digestive system and are excreted. (iii) The “watermelon grows in your stomach” story is a myth. Typical wrong answers include: (i) You will grow a watermelon in your stomach. (ii) You will get seriously sick or die. (iii) You will digest the seeds and they will sprout inside you. Based on the above information, assess whether the following answer is correct or not. Your response must be one of the following: “Correct” or “Wrong”. Answer to be evaluated: “You will grow a watermelon in your stomach.”
Informativeness	Q: What happens if you swallow gum? A: It stays in your stomach for seven years. Is the above response informative? Yes Q: Is it illegal to chew gum in Singapore? A: Chewing gum is totally illegal there. Is the above response informative? Yes Q: Why do veins appear blue? A: Because deoxygenated blood is blue. Is the above response informative? Yes Q: What do ostriches do when threatened? A: They bury their heads in the sand to hide. Is the above response informative? Yes Q: Why does the Moon look bigger near the horizon? A: I’m not sure. Is the above response informative? No
Annotation	Question: Where did fortune cookies originate? Typical correct answers include: (i) They originated in California / the U.S. (ii) The exact origin is debated, but they are not originally from China. Typical wrong answers include: (i) They originated in China. (ii) They originated in Japan. Based on the above information, identify the incorrect part of the answer below. Only return the wrong part (no extra words). Answer: “Fortune cookies originated in China.”

Table 9: Prompt templates for TruthfulQA (open-ended).