CAN GRAY-BOX RIVAL WHITE-BOX? ANTICIPATORY CONSISTENCY PROPAGATION WITH TOP-K CANDIDATES FOR LLM HALLUCINATION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucinations in large language models are a major obstacle to their applications. While white-box detectors achieve strong performance by exploiting internal model states, such information is inaccessible for closed-source LLMs. Gray-box methods, on the other hand, offer a practical alternative by leveraging only API-accessible Top-K candidates and their probabilities, but their performance remains limited. In this paper, we seek to bridge this gap by introducing rich external semantic representations into gray-box detectors. However, since external embedding models and the target LLM are trained independently, an inherent information gap could exist between the external semantic space and the target model's generation dynamics. To address this, we propose **Prop-K**, a consistency-guided propagation method for gray-box hallucination detection. Prop-K captures the predictive consistency of the target model's generation process at two complementary levels: **local agreement consistency** measures how Top-K candidates support or conflict with the current Top-1 prediction at each step, while **global anticipatory consistency** captures whether early predictive expectations align with or deviate from what the model generates later. Together, these two consistency signals guide semantic propagation throughout the generated sequence, potentially aligning external semantic embeddings with the target model's predictive behavior. Empirically, Prop-K consistently outperforms existing gray-box methods with an average gain in AUROC of 0.04–0.07 **across 7 target LLMs and 4 benchmarks**. Notably, despite having no access to internal model states, Prop-K matches and, in some cases, even surpasses the strongest white-box detector with such privileged access.

1 INTRODUCTION

Accurate hallucination detection is essential for reducing the risks of LLM deployment in critical application domains. Existing approaches fall into two categories. *White-box methods* exploit internal states of the target LLM, such as hidden representations and attention patterns, and typically achieve state-of-the-art performance (Niu et al., 2025; Frasca et al., 2026; Orgad et al., 2025), but they are impractical for closed-source commercial models. *Black-box methods*, in contrast, rely only on generated text (Manakul et al., 2023; Farquhar et al., 2024) and are broadly deployable, but their performance is limited by the lack of fine-grained information about the generation process.

Recently, **gray-box methods** have emerged as an attractive trade-off, leveraging API-accessible Top-K candidates and their probabilities without accessing model internals (Bar-Shalom et al., 2026; Shapiro et al., 2026). Yet their performance remains limited because fine-grained token representations are unavailable, forcing existing methods to rely largely on low-dimensional statistics such as probabilities and entropy. As a result, semantically different generation states with similar probability profiles can be difficult to distinguish. This raises a key question – *can richer semantic and structural signals be recovered from limited gray-box observations, so as to narrow the gap between white-box and gray-box detection?*

A natural idea is to enrich gray-box signals with external token semantics (embeddings). However, combining external embeddings with probability-based features yields only negligible gains (see Motivating Experiments). We attribute this to a fundamental challenge: in white-box settings,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

semantic representations and generation probabilities are from the same model and naturally coupled; in gray-box settings, by contrast, external embeddings are produced by an independent model and are therefore unaligned with the target LLM's predictive behavior. We believe that bridging this gap requires more than fusion of the two information sources. Instead, it needs a more principled interaction mechanism that can reshape and calibrate external semantic representations to better reflect the target model's predictive structure and probabilistic preferences.

Building on this observation, we propose **Prop-K**, a consistency-guided propagation method for gray-box hallucination detection. Rather than treating generation probabilities as auxiliary confidence features, Prop-K jointly exploits predictive probabilities and external semantics to uncover structured consistency signals at two levels: **local agreement consistency** captures how Top-K candidates support or conflict with the Top-1 prediction, while **global anticipatory consistency** measures whether early predictive preferences align with or deviate from subsequent generations. Together, these signals guide semantic propagation across the sequence, calibrating external representations toward the target model's predictive behavior for more reliable detection.

As a gray-box method, Prop-K consistently outperforms gray-box baselines, improving the average AUROC by about 0.07 across 16 model–dataset settings (0.783 to 0.849). More notably, despite having no access to any privileged internal states of the target model, Prop-K matches (and even slightly surpasses) the strongest white-box method on average (0.849 vs. 0.847).

Our contributions are summarized as follows:

1. We introduce external token embeddings into gray-box detection to enrich its input signals. We also show that the gap between external semantics and the target model's predictive dynamics is a key factor limiting the effectiveness of their direct fusion.
2. We propose consistency-guided propagation to calibrate external token semantics with the target model's predictive behavior, together with a dual-anchor semantic–temporal module to learn more effective representations for hallucination detection.
3. We conduct extensive experiments to show that Prop-K effectively improves gray-box detection and achieves a similar level of performance compared with white-box methods.

2 Related Work

Existing hallucination detection methods can be classified based on the model signals accessed. Black-box methods use only the text generated by the target model. For example, SelfCheckGPT assesses agreement across multiple sampled responses (Manakul et al., 2023). Semantic Entropy estimates uncertainty from the frequencies of semantic clusters (Farquhar et al., 2024). OpenFactCheck verifies factual claims using external evidence (Iqbal et al., 2024). White-box methods can access internal signals from the target model. For example, Orgad et al. (2025) predict answer correctness from the hidden states of core answer spans. HaMI uses uncertainty to enhance hidden representations and adaptively selects key tokens through multiple-instance learning (Niu et al., 2025). CHARM detects hallucinations through message passing on attention graphs (Frasca et al., 2026). As a trade-off between the two, gray-box methods primarily use observable generation probabilities as detection signals. For example, EPR estimates uncertainty by averaging predictive entropy across generation steps (Moslonka et al., 2025). LOS-Net uses a Transformer to capture sequential patterns in candidate token probabilities and the ranks of generated tokens (Bar-Shalom et al., 2026). HALT models the temporal dynamics of candidate token log probabilities and their statistical features using a bidirectional GRU (Shapiro et al., 2026).

Prior work has shown that LLMs often encode signals about downstream outputs early in the generation process. For example, early hidden states have been shown to encode information about hallucination risk (Snyder et al., 2024; Gottesman & Geva, 2024), answer confidence and output format (Ashok & May, 2025), response length (Dong et al., 2025; Xie et al., 2026). More recently, Liu et al. (2026) show that such information is also reflected directly in generation probabilities: *the softmax distribution at each step can capture not only the immediate next-token preference but also the composition of tokens farther into the future.* These findings motivate us to examine whether early predictive preferences remain consistent with subsequent generation, which we formulate as *anticipatory consistency* for richer gray-box pattern modeling.

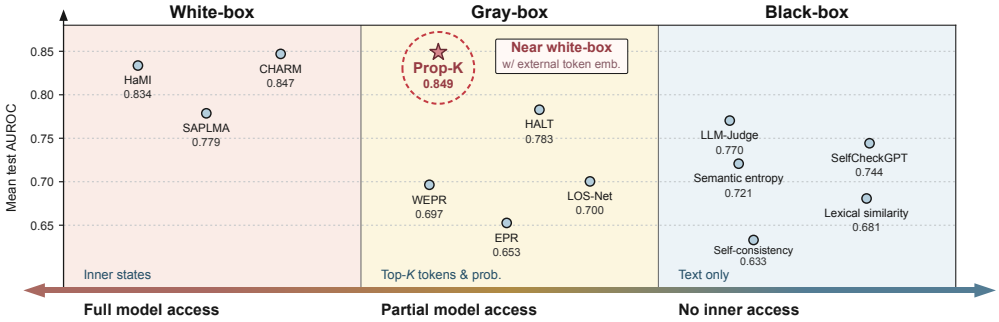


Figure 1: Average AUROC of white-box, gray-box, and black-box methods (across 4 open-source models and 4 datasets, each over three seeds), more results in Table 1. The proposed Prop-K method notably narrows the gap between white-box and gray-box methods.

3 METHODS

Motivating Experiments. We first examine whether external semantics can directly improve gray-box detection. *(1) Direct feature integration.* We augment HALT (Shapiro et al., 2026), a gray-box method based solely on probability-derived features, with external token embeddings. Using either Top-1 or Top-20 candidates, the AUROC improvement on HaluEval and GSM8K is negligible and below 0.01 (Appendix D.6). *(2) Probability-modulated interaction.* Inspired by the white-box method HaMI (Niu et al., 2025), we further use inner generation probabilities to modulate external token embeddings - a tighter probability–semantic interaction. Yet, replacing HaMI's internal embeddings with external ones reduces AUROC by 0.09 and 0.07 on HaluEval and GSM8K, respectively.

These results suggest that simply combining external semantics with generation probabilities is insufficient. We argue that the key bottleneck lies in the gap between external semantic representations and the target LLM's own predictive behavior, which arises from two independent models. We next show how this gap can be reduced using "predictive consistency signals" jointly characterized by these two information sources.

3.1 PROBLEM SETUP

Given an input prompt x, the target LLM generates a response $y = (y_1, \ldots, y_L)$, where L is the number of generation steps. Under the gray-box setting, at each step t, we have access to the Top-K candidate tokens with their generation probabilities, and an external semantic embedding model $\mathbb{E}$. Our goal is to determine whether the generated answer sequence contains hallucinations.

We assume that Top-K candidates are ranked in descending order of probability, and y_t is the Top-1 candidate at step t. Then the information available at *each generation step* t is divided into two parts:

- **Selected Token.** The Top-1 token y_t, with external semantic representation $\mathbf{y}_t = \mathbb{E}(y_t)$, and generation probability $p_t(y_t) = P_\theta(y_t \mid x, y_{<t})$.
- **Alternative Candidates.** The remaining Top-K candidates $\mathcal{C}_t = \{c_{t,k}\}_{k=2}^{K}$, with external embedding $\mathbf{c}_{t,k} = \mathbb{E}(c_{t,k})$, generation probability $p_t(c_{t,k}) = P_\theta(c_{t,k} \mid x, y_{<t})$, for $_{k=2,\ldots,K.}$

3.2 PROP-K

As illustrated in Figure 2, Prop-K consists of three components. (1) **Answer Sequence Encoding** obtains the Top-K candidates and their generation probabilities from the target LLM, together with external token embeddings. (2) **Consistency-Guided Propagation** captures both *local agreement consistency* (for each generation step) and *global anticipatory consistency* (across multiple generation steps), and uses these consistency signals to guide the propagation and calibration of external semantic representations. (3) **Dual-Anchor Readout** summarizes the refined token representations along semantic and temporal dimensions to extract key features for answer-level hallucination detection.

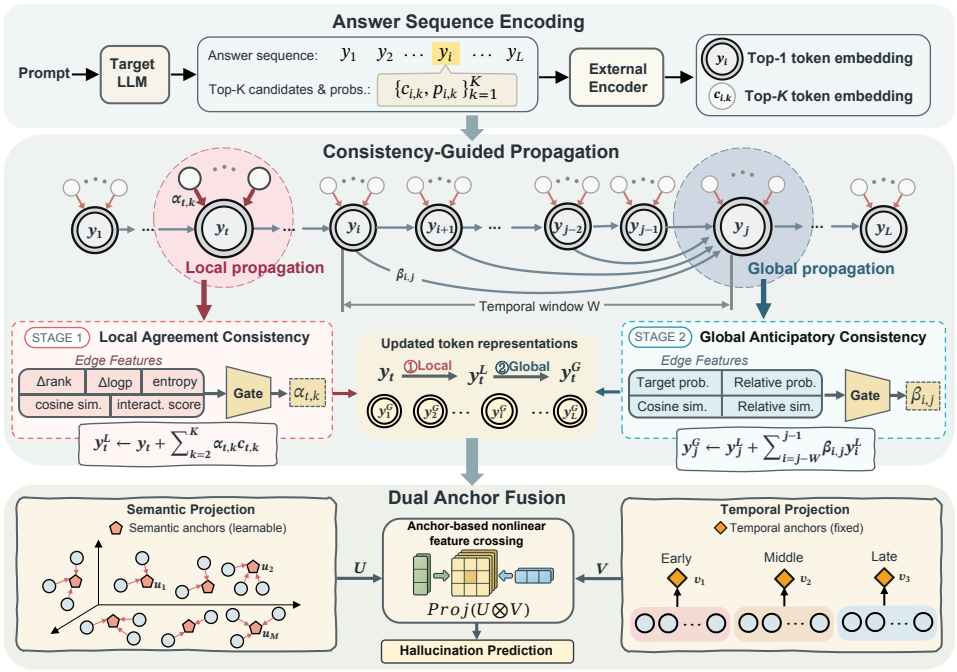


Figure 2: Workflow of Prop-K (sequence encoding, consistency propagation, and dual-anchor fusion).

3.2.1 Local Consistency Propagation

Local agreement consistency (LAC) measures how strongly the alternative Top-K candidates support the selected Top-1 token. It is high when semantically compatible candidates also receive substantial probability mass, and low when high-probability alternative tokens are irrelevant to the Top-1 token. This signal guides selective propagation of candidate semantics, enabling the Top-1 representation to better aggregate step-wise agreement and reflect the target model's local predictive preference.

Local Agreement Consistency Graph $\mathcal{G}_t^L$. For each generation step t, we construct a directed graph $\mathcal{G}_t^L$, with the Top-1 token y_t as the center, and other candidates $c_{t,k}$ $(k = 2, \ldots, K)$ as peripheral nodes. Each candidate $c_{t,k}$ has a directed edge $c_{t,k} \to y_t$ to the center, with no links among candidates. An external model then embeds the center and candidate nodes as $\mathbf{y}_t$ and $\mathbf{c}_{t,k}$, respectively.

Candidate semantics propagate along these edges and integrate into the center via a residual update:

$$\mathbf{y}_t^{\mathrm{L}} = \mathrm{LN}\Big(\phi(\mathbf{y}_t) + \textstyle\sum_{k=2}^{K} \alpha_{t,k}\, \phi(\mathbf{c}_{t,k})\Big)\,. \tag{1}$$

Here, ϕ is a learnable transform, and LN is layer normalization. The gating weight $\alpha_{t,k}$ controls how much information candidate $c_{t,k}$ propagates to the center node y_t.

We design the following local agreement cues to compute the gating weights $\alpha_{t,k}$:

(1) Semantic Relatedness. We measure the semantic relatedness between the k-th candidate and the Top-1 token using cosine similarity:

$$s_{t,k} = \cos(\mathbf{c}_{t,k}, \mathbf{y}_t)\,. \tag{2}$$

A larger $s_{t,k}$ indicates that the k-th candidate is more semantically related to the Top-1 token and is expected to provide stronger support for it.

(2) Rank Distance. We measure the difference in predictive ranking between the k-th candidate and the Top-1 token using logarithmic rank distance:

$$d_{t,k}^{\mathrm{r}} = \big|\log\big[\mathrm{rank}_t(c_{t,k})\big] - \log\big[\mathrm{rank}_t(y_t)\big]\big|\,. \tag{3}$$

(3) Log-Probability Gap. We compute the log-probability difference between the k-th candidate and the Top-1 token as:

$$d_{t,k}^{\mathrm{p}} = \log p_t(y_t) - \log p_t(c_{t,k}). \tag{4}$$

For both the rank distance $d^{\mathrm{r}}_{t,k}$ and the log-probability gap $d^{\mathrm{p}}_{t,k}$, smaller values indicate that the k-th candidate is more strongly preferred by the model. Together with the semantic similarity $s_{t,k}$, these signals characterize how strongly the candidate supports or disagrees with the Top-1 decision.

(4) Top-K Entropy. Let $c_{t,1} = y_t$ denote the Top-1 token. We renormalize the Top-K probabilities to obtain $\bar{p}_t(c_{t,k})$ and compute their Shannon entropy:

$$e_t = -\sum_{k=1}^{K} \bar{p}_t(c_{t,k}) \log \bar{p}_t(c_{t,k}). \tag{5}$$

A larger e_t indicates greater uncertainty at the current step and potentially weaker support for the Top-1 token.

Finally, these features are fed into a unified gating network to compute the weight $\alpha_{t,k}$:

$$\alpha_{t,k} = \mathrm{softmax}_k\Big(\mathrm{MLP}\Big([d^{\mathrm{r}}_{t,k}; d^{\mathrm{p}}_{t,k}; e_t; s_{t,k}; \eta_{t,k}]\Big)\Big). \tag{6}$$

Empirically, we also include a lightweight compatibility term that maps each Top-1–candidate pair, $(y_t, c_{t,k})$, to a scalar score $\eta_{t,k}$, providing a flexible semantic adjustment complementary to the explicit consistency cues. Details are provided in Appendix B.

3.2.2 Longer-Range Consistency Propagation

Global anticipatory consistency (GAC) captures whether earlier predictive preferences remain consistent with tokens generated several steps later. As discussed in Section 2, **step-wise softmax distributions can encode information beyond the immediate next token**. If subsequently generated content receives strong probabilistic and semantic support from multiple earlier steps, it is consistent with the model's prior predictions; otherwise, it indicates cross-step inconsistency or predictive drift. Based on this, we construct a probability-grounded anticipatory graph to guide long-range semantic propagation, injecting global consistency signals into external semantic representations.

Global Anticipatory Consistency Graph $\mathcal{G}^{\mathrm{G}}_j$. For each generation step j, we construct a directed graph $\mathcal{G}^{\mathrm{G}}_j$ containing the current Top-1 token y_j and the Top-1 tokens from the preceding W steps. For any $i < j$ in this window, we add an edge $y_i \to y_j$ if y_j appeared among the Top-K candidates at step i, indicating anticipatory support from step i for the later generation at step j.

We then characterize the consistency strength of each cross-step link using the following features:

(1) Anticipatory Probability. For the Top-1 token y_j generated at step j, we retrieve the probability assigned to it in the softmax distribution p_i at the earlier step i:

$$p_{j|i} = p_i(y_j). \tag{7}$$

A larger $p_{j|i}$ indicates stronger anticipatory probabilistic support from earlier step i to later step j.

(2) Relative Anticipatory Probability. The anticipatory probability mass from step i may also be distributed among the unselected candidate tokens in $\mathcal{C}_j$ at the subsequent step j. We therefore measure how strongly support for y_j stands out relative to the other candidates:

$$r_{j|i} = \frac{p_i(y_j)}{p_i(y_j) + \sum_{c \in \mathcal{C}_j} p_i(c)}. \tag{8}$$

A larger $r_{j|i}$ indicates that the predictive support from step i is more concentrated on the Top-1 token y_j, corresponding to stronger cross-step predictive consistency.

(3) Semantic Relatedness. We measure semantic correlation of the Top-1 tokens at steps i and j:

$$\chi_{i,j} = \cos(\mathbf{y}_i, \mathbf{y}_j). \tag{9}$$

A larger $\chi_{i,j}$ indicates greater semantic relatedness between y_i and y_j, reflecting stronger semantic support from the earlier step i for the subsequent Top-1 token y_j.

(4) Relative Semantic Relatedness. Similarly, y_i may also be semantically related to the other candidate tokens $\mathcal{C}_j$ at step j. To measure how strongly the semantic support from y_i for the Top-1 token y_j stands out relative to the other candidates, we define:

$$u_{i,j} = \frac{\exp\left(\chi_{i,j}/\tau\right)}{\exp\left(\chi_{i,j}/\tau\right) + \sum_{k=2}^{K} \exp\left(\cos\left(\mathbf{y}_i, \mathbf{c}_{j,k}\right)/\tau\right)}, \tag{10}$$

where τ is temperature. A larger $u_{i,j}$ means stronger, more concentrated anticipatory support from y_i towards the selected token y_j at step j.

Together, these features capture both probabilistic and semantic consistency between step i and step j, which are then fed into a gating module to compute the propagation weight:

$$\beta_{i,j} = \sigma\left(\mathrm{MLP}\left([p_{j|i};\, r_{j|i};\, \chi_{i,j};\, u_{i,j}]\right)\right). \tag{11}$$

Then the information propagation guided by this global anticipatory consistency is defined as:

$$\mathbf{y}_j^{\mathrm{G}} = \mathrm{LN}\left(\mathbf{y}_j^{\mathrm{L}} + \tfrac{1}{d_j}\textstyle\sum_{i:\, y_i \to y_j} \beta_{i,j}\mathbf{y}_i^{\mathrm{L}}\right). \tag{12}$$

Here, LN is layer normalization, and d_j is the in-degree of the node y_j. For nodes with no incoming edges, the aggregation term is set to zero.

The multi-step consistency graph $\mathcal{G}^{\mathrm{G}}$ captures a very different relation from causal attention. Attention models **contextual dependence**, i.e., how preceding tokens contribute to the current prediction, whereas our *anticipatory consistency graph* models **expectation realization**, i.e., whether predictive expectations formed at earlier steps are subsequently supported or violated by later generations. This cross-step consistency provides a valuable signal for hallucination detection, and our ablation study (Table 5) further confirms its greater effectiveness in improving detection performance.

Eqs. (1) and (12) take the form of graph propagation. However, rather than being generic neighborhood aggregation, the information flow is guided by predictive consistency across candidates and generation steps. The gating functions regulate how probabilistic/semantic cues are used to control the amount of information propagated, enabling flexible adaptation to the target model's behavior.

3.2.3 Semantic and Temporal Anchor Readout

Having obtained the refined sequence representations $\mathbf{Y}^{\mathrm{G}} = (\mathbf{y}_1^{\mathrm{G}}, \ldots, \mathbf{y}_L^{\mathrm{G}})$ (of variable lengths), we next compress it into a fixed-dimensional representation. To reduce the information loss of naive pooling, we project the token-level representations onto two complementary sets of anchors: **semantic anchors** summarize representative semantic patterns across the sequence, while **temporal anchors** preserve stage-specific information. The two sets of representations are then subject to nonlinear interaction and weighted aggregation to obtain the final feature.

Semantic anchors are a set of learnable prototypes $\{\mathbf{z}_m\}_{m=1}^{M}$ to cover key semantic patterns in $\mathbf{Y}^{\mathrm{G}} = (\mathbf{y}_1^{\mathrm{G}}, \ldots, \mathbf{y}_L^{\mathrm{G}})$ - they are initialized through clustering (Appendix B.3) and optimized end-to-end. To aggregate token-level embeddings into these anchors, we measure the similarity between each token and the semantic anchors, and obtain the representation for the m-th anchor as:

$$\mathbf{u}_m = \frac{\sum_{t=1}^{L} \exp\big(\langle \mathbf{y}_t^{\mathrm{G}}, \mathbf{z}_m\rangle/\tau'\big)\, \mathbf{y}_t^{\mathrm{G}}}{\sum_{t=1}^{L} \exp\big(\langle \mathbf{y}_t^{\mathrm{G}}, \mathbf{z}_m\rangle/\tau'\big)}, \qquad m = 1, \ldots, M, \tag{13}$$

where τ' is temperature. This yields a set of M semantic anchor-level representations $U = \{\mathbf{u}_m\}_{m=1}^{M}$.

Temporal anchors summarize stage-specific representation patterns in $\mathbf{Y}^{\mathrm{G}} = (\mathbf{y}_1^{\mathrm{G}}, \ldots, \mathbf{y}_L^{\mathrm{G}})$. We uniformly partition each variable-length response into $N = 3$ stages (early, middle, and late) and aggregate token embeddings within each stage. Let $\mathcal{I}_n$ be the token indices in stage-n. The aggregated feature for the n-th temporal anchor is:

$$\mathbf{v}_n = \frac{\sum_{t\in\mathcal{I}_n} \mathrm{Att}_\theta(\mathbf{y}_t^{\mathrm{G}})\mathbf{y}_t^{\mathrm{G}}}{\sum_{t\in\mathcal{I}_n} \mathrm{Att}_\theta(\mathbf{y}_t^{\mathrm{G}})}, \qquad n = 1, \ldots, N, \tag{14}$$

with $\mathrm{Att}_\theta(\cdot)$ a learnable scoring function that quantifies token contributions. This yields a compact set of N temporal anchor representations, denoted as $V = \{\mathbf{v}_n\}_{n=1}^{N}$.

Dual-Anchor Fusion. Given the anchor-level features from the semantic anchors $U = \{\mathbf{u}_m\}_{m=1}^{M}$ and temporal anchors $V = \{\mathbf{v}_n\}_{n=1}^{N}$, we model their pairwise interactions (joint semantic–temporal structure) through outer product, and then perform weighted combination, as:

$$\mathbf{R} = \textstyle\sum_{m=1}^{M}\sum_{n=1}^{N} f(\mathbf{u}_m)\, g(\mathbf{v}_n)\, \big[\psi(\mathbf{u}_m) \otimes \psi'(\mathbf{v}_n)\big], \tag{15}$$

where $f(\mathbf{u}_m)$ and $g(\mathbf{v}_n)$ are MLP-based weighting functions to quantify the importance of the anchors $\mathbf{u}_m$ and $\mathbf{v}_n$, respectively, while ψ and ψ' are learnable projections (Appendix B.3). The integrated feature $\mathbf{R}$ will be sent to a classifier with binary cross-entropy loss to train the model.

Table 1: Test AUROC (↑) across four datasets and four open-source target LLMs. Each result is averaged over **three random seeds**; standard deviations are reported in Table 9 (Appendix D.1).

	Mistral-7B					Llama3-8B				
Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Black-box methods										
Self-consistency	0.595	0.583	0.579	0.762	0.630	0.664	0.652	0.597	0.820	0.683
Lexical similarity	0.649	0.701	0.704	0.772	0.707	0.675	0.652	0.661	0.822	0.703
LLM-Judge (GPT-4o)	0.853	0.716	0.727	0.817	0.778	0.809	0.799	0.713	0.749	0.768
SelfCheckGPT-NLI	0.696	0.689	0.684	0.833	0.726	0.673	0.728	0.708	0.907	0.754
Semantic Entropy	0.711	0.712	0.712	0.748	0.720	0.706	0.740	0.723	0.815	0.746
Gray-box methods										
EPR	0.649	0.644	0.655	0.586	0.634	0.710	0.669	0.676	0.632	0.672
WEPR	0.656	0.668	0.670	0.659	0.663	0.732	0.735	0.711	0.713	0.723
LOS-Net	0.676	0.691	0.664	0.657	0.672	0.723	0.670	0.656	0.721	0.693
HALT	0.779	0.751	0.762	0.750	0.761	0.788	0.720	0.748	0.816	0.768
White-box methods										
SAPLMA	0.802	0.753	0.742	0.786	0.771	0.780	0.828	0.794	0.786	0.797
HaMI	**0.875**	0.788	**0.818**	0.829	0.828	0.839	0.818	0.795	0.881	0.833
CHARM	0.864	**0.811**	0.812	0.843	**0.833**	0.858	**0.836**	**0.817**	0.887	**0.850**
Prop-K	0.846	0.808	0.808	**0.855**	0.829	**0.863**	0.824	0.801	**0.913**	**0.850**

	Gemma3-4B					Gemma3-12B				
Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Black-box methods										
Self-consistency	0.616	0.585	0.585	0.740	0.632	0.590	0.505	0.515	0.740	0.587
Lexical similarity	0.658	0.643	0.669	0.747	0.679	0.636	0.568	0.591	0.744	0.635
LLM-Judge (GPT-4o)	**0.903**	0.689	0.667	0.817	0.769	0.862	0.791	0.771	0.642	0.767
SelfCheckGPT-NLI	0.785	0.748	0.736	0.773	0.760	0.795	0.709	0.700	0.744	0.737
Semantic Entropy	0.743	0.731	0.743	0.732	0.737	0.726	0.620	0.638	0.732	0.679
Gray-box methods										
EPR	0.649	0.642	0.665	0.682	0.659	0.612	0.648	0.661	0.662	0.646
WEPR	0.673	0.678	0.709	0.797	0.714	0.680	0.667	0.673	0.723	0.686
LOS-Net	0.740	0.744	0.755	0.700	0.735	0.733	0.713	0.717	0.646	0.702
HALT	0.810	0.802	0.806	0.842	0.815	0.794	0.766	0.782	0.809	0.788
White-box methods										
SAPLMA	0.822	0.731	0.752	0.817	0.781	0.852	0.809	0.795	0.610	0.767
HaMI	0.856	0.790	0.789	0.910	0.836	0.853	0.814	0.804	**0.879**	0.838
CHARM	0.877	0.836	0.820	0.890	0.856	**0.882**	**0.846**	**0.827**	0.846	**0.850**
Prop-K	0.846	**0.850**	**0.853**	**0.928**	**0.869**	0.872	**0.846**	0.822	0.848	0.847

Unlike direct pooling or pre-selecting key tokens, our approach avoids irreversibly mixing or discarding information at an early stage. Instead, it first preserves relatively fine-grained semantic and temporal representations, and then lets the classifier determine which patterns, stages, and combinations are most useful, and thus could more adaptively preserve useful evidence of hallucination.

4 EXPERIMENTS

Datasets. We evaluate answer-level hallucination detection on four benchmarks: TriviaQA (Joshi et al., 2017) and HaluEval (Li et al., 2023a) for factual question answering, HotpotQA (Yang et al., 2018) for factual reasoning, and GSM8K (Cobbe et al., 2021) for math.

Baselines and Metrics. We include representative methods from all three categories: (1) **black-box** methods, including SelfCheckGPT-NLI (Manakul et al., 2023) and Semantic Entropy (Farquhar et al., 2024), both using 20 rollouts from the target model; (2) **gray-box** methods, including EPR (Moslonka et al., 2025), WEPR, HALT (Shapiro et al., 2026), and LOS-Net (Bar-Shalom et al., 2026); and (3) **white-box** methods, including SAPLMA (Azaria & Mitchell, 2023), HaMI (Niu et al., 2025), and CHARM (Frasca et al., 2026). We report answer-level AUROC on the test set. Following Luo et al. (2025), DeepSeek-V4-Flash (DeepSeek-AI et al., 2026) serves as a common judge for answer correctness based on the question, reference answer, and generated response (Appendix C.1).

All methods evaluate the same generated answer under a shared prompt. White-box methods access internal states of open-source target models, black-box methods use only the generated text, and gray-box methods use the Top-20 candidate tokens and their log-probabilities; Prop-K additionally uses

Table 2: Test AUROC (↑) on four datasets and three closed-source target LLMs. Each result is averaged over **three random seeds**; standard deviations are reported in Table 10 (Appendix D.1).

Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
GPT-4.1-mini					
LLM-Judge (GPT-4o)	0.827	0.732	0.710	0.567	0.709
SelfCheckGPT-NLI	0.728	0.796	0.765	<u>0.787</u>	0.769
Semantic Entropy	0.800	0.801	0.758	0.714	0.768
LOS-Net	0.722	0.754	0.713	0.734	0.731
HALT	<u>0.848</u>	<u>0.821</u>	<u>0.796</u>	0.735	<u>0.800</u>
Prop-K	**0.878**	**0.844**	**0.823**	**0.826**	**0.843**
Gemini-2.5-Flash-Lite					
LLM-Judge (GPT-4o)	0.781	0.598	0.774	0.587	0.685
SelfCheckGPT-NLI	**0.876**	0.712	0.816	<u>0.860</u>	0.816
Semantic Entropy	0.776	0.532	0.665	0.608	0.645
LOS-Net	0.794	0.854	0.743	0.838	0.807
HALT	0.854	<u>0.856</u>	<u>0.820</u>	0.853	<u>0.846</u>
Prop-K	<u>0.868</u>	**0.873**	**0.838**	**0.936**	**0.879**
DeepSeek-V4-Flash					
LLM-Judge (GPT-4o)	0.731	0.623	0.768	0.519	0.660
SelfCheckGPT-NLI	<u>0.825</u>	0.556	0.728	0.767	0.719
Semantic Entropy	0.779	0.543	0.694	0.711	0.682
LOS-Net	0.714	0.837	0.775	0.755	0.770
HALT	0.774	<u>0.852</u>	<u>0.790</u>	<u>0.782</u>	<u>0.799</u>
Prop-K	**0.852**	**0.874**	**0.833**	**0.836**	**0.849**

external token embeddings. We use greedy decoding for open-source models and low-randomness API settings for closed-source models to reduce generation variance and ensure stable single-response comparisons (Vashurin et al., 2025; Aichberger et al., 2026). Prompt templates in Appendix C.2.

Target Models and Comparison Settings. We report results under two complementary settings:

- **Open-Source Target Models.** We evaluate Mistral-7B (Jiang et al., 2023), Llama3-8B (Grattafiori et al., 2024), Gemma3-4B, and Gemma3-12B (Team et al., 2025) with black-box, gray-box, and white-box detectors (gray-/black-box access is additionally imposed on these open models for comprehensive comparisons).
- **Closed-Source Target Models.** We evaluate GPT-4.1-mini (OpenAI), Gemini-2.5-Flash-Lite (Comanici et al., 2025), and DeepSeek-V4-Flash with black-box and gray-box methods.

Implementation. Prop-K uses a frozen `all-MiniLM-L6-v2` encoder (Wang et al., 2020) for 384-dimensional token embeddings. We train with AdamW (Loshchilov & Hutter, 2019) (batch size 32, up to 20 epochs). We fix the hidden dimension at $d = 128$ and use $M = 16$ semantic anchors. We tune three architecture hyperparameters (the numbers of local and global propagation rounds and the global window size W), with two candidate values for each. Both hyperparameter and checkpoint selection are based on validation AUROC. More details are in Appendix C.3.

4.1 Main Results

We conduct comprehensive evaluations on four datasets and seven LLMs (four open-source and three closed-source). To reduce statistical fluctuation, each result is the mean AUROC over **three random seeds**, with standard deviations in Appendix D.1. We use two-sided paired Student's t-tests to compare Prop-K with each baseline. For consistency-based black-box methods, which already rely on repeated stochastic sampling, we report a single run due to their high computational/token costs.

Open-Source Target Models. Table 1 reports results of all three classes of detection methods on four open-source LLMs and four datasets. Compared with the strongest **gray-box baseline**, HALT, Prop-K improves the average AUROC by 0.05–0.08. It also outperforms **black-box methods** by 0.05–0.10 in average AUROC, while avoiding their need for repeated answer sampling. Two-sided paired Student's t-tests further confirm that the improvements over all evaluated black-box and gray-box baselines are statistically significant ($p < 0.01$); detailed results are in Appendix D.2.

Prop-K achieves performance essentially on par with the strongest **white-box detector**, CHARM, while attaining a slightly higher average AUROC (0.849 vs. 0.847) across 16 model–dataset pairs. The difference is not statistically significant ($p = 0.687$; Appendix D.2), indicating that Prop-K effectively reaches white-box-level detection performance without access to internal model states.

Closed-Source Target Models. Table 2 reports results of gray- and black-box methods on closed-source target models. Prop-K has a consistent advantage with an average AUROC of 0.857 over 12 model–dataset pairs, while the strongest gray-box baseline is at 0.815. Overall, Prop-K performance gains are consistent across different benchmark datasets and target-LLMs.

4.2 Ablation Studies

We report three-seed mean AUROC for ablations on HaluEval and GSM8K using Gemma3-12B, with standard deviations in Table 11 of Appendix D.1.

Table 3: Consistency-guided propagation.

Configuration	HaluEval	GSM8K
w/o Local	0.791	0.798
w/o Global	0.827	0.821
w/o Both	0.714	0.726
Prop-K	**0.846**	**0.848**

Table 4: External embedding models.

Embedding	HaluEval	GSM8K
Jina	0.843	0.842
E5	0.848	0.835
GTE	0.843	0.857
MiniLM (default)	**0.846**	**0.848**

Table 5: Global propagation variants.

Configuration	HaluEval	GSM8K
w/o Global	0.827	0.821
Self-attention	0.830	0.834
Prop-K ($W = 1$)	0.833	0.812
Prop-K ($W \gg 1$)	**0.846**	**0.848**

Table 6: Readout methods.

Readout	HaluEval	GSM8K
Mean pooling	0.813	0.812
Max pooling	0.831	0.832
Attention pooling	0.824	0.822
Dual-anchor	**0.846**	**0.848**

How critical is consistency propagation to Prop-K's performance? As shown in Table 3, removing the entire consistency propagation module causes a serious degradation, reducing AUROC by about 0.13. Removing local consistency alone leads to a 0.05 drop, while removing global consistency results in a roughly 0.03 decrease. These results indicate that local and global consistency capture complementary information, and that their joint modeling is critical to Prop-K's overall performance.

Is Prop-K sensitive to the choice of external embedding model? As shown in Table 4, Prop-K remains stable across different external embedding models, with only minor AUROC variations (less than 0.01 on HaluEval and about 0.01 on GSM8K), thus being largely insensitive to specific encoders. The exact model identifiers are provided in Appendix C.3.

How do different forms of longer-range propagation affect results? As shown in Table 5, replacing longer-range consistency with parameter-matched self-attention reduces AUROC by about 0.02, while reducing the window size to 1 leads to an AUROC drop of about 0.03. These results demonstrate that Prop-K benefits not only from long-range interactions, but also from the specific "anticipatory connectivity" between early predictions and subsequent outputs.

Is dual-anchor readout more useful than conventional pooling? As in Table 6, replacing dual-anchor readout with token-level mean pooling, max pooling, or content-based attention pooling on the consistency-enhanced sequence $\mathbf{Y}^{\mathrm{G}}$ leads to an AUROC drop by 0.02–0.04. These demonstrate the effectiveness of the dual-anchor readout module in preserving useful semantic and temporal details.

5 Conclusion

In this paper, we propose Prop-K, extending conventional gray-box hallucination detection by introducing rich external semantic representations. To overcome the resulting information gap between external semantic spaces and the target LLM's generation behavior, Prop-K leverages local agreement consistency and global anticipatory consistency to guide semantic propagation and align the two sources of information. Extensive experiments show consistent improvements over existing gray-box methods, with performance matching or even surpassing s.o.t.a. white-box detectors.

REFERENCES

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking uncertainty estimation in LLMs: A principled single-sequence measure. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wdhruVcRx1.

Dhananjay Ashok and Jonathan May. Language models can predict their own behavior. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2102. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/5a9c1af5f76da0bd37903b6f23e96c74-Abstract-Conference.html.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Guy Bar-Shalom, Fabrizio Frasca, Derek Lim, Yoav Gelberg, Yftah Ziser, Ran El-Yaniv, Gal Chechik, and Haggai Maron. Beyond next token probabilities: Learnable, fast detection of hallucinations and data contamination on LLM output distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30058–30066, 2026. doi: 10.1609/aaai.v40i36.40254. URL https://ojs.aaai.org/index.php/AAAI/article/view/40254.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14301–14320. PMLR, 2025. URL https://proceedings.mlr.press/v267/dong25p.html.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://www.nature.com/articles/s41586-024-07421-0.

Fabrizio Frasca, Guy Bar-Shalom, Yftah Ziser, and Haggai Maron. Neural message-passing on attention graphs for hallucination detection. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/014898b821e3ce966ae1db68cb8d9966-Abstract-Conference.html.

Daniela Gottesman and Mor Geva. Estimating knowledge in large language models without generating a single token. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3994–4019. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.232. URL https://aclanthology.org/2024.emnlp-main.232/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Nenkov Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. OpenFactCheck: A unified framework for factuality evaluation of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 219–229. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-demo.23. URL https://aclanthology.org/2024.emnlp-demo.23/.

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. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6449–6464. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.emnlp-main.397. URL https://aclanthology.org/2023.emnlp-main.397/.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning, 2023b. URL https://arxiv.org/abs/2308.03281.

Yihe Liu, Huibin Wang, Xianming Hu, Pinyi Zhang, Jiahao Xiong, Chenglin Wang, Nuoyi Chen, Hongbo Zhao, Jie Zhang, and Kai Zhang. Next-ToBE: Probabilistic next token-bag exploitation for activating anticipatory capacity in LLMs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8IJojfaOh.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jinyuan Luo, Zhen Fang, Yixuan Li, Seongheon Park, and Ling Chen. Shaking to reveal: Perturbation-based detection of llm hallucinations, 2025. URL https://arxiv.org/abs/2506.02696.

Potsawee Manakul, Adian Liusie, and Mark Gales. 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*, pp. 9004–9017. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Charles Moslonka, Hicham Randrianarivo, Arthur Garnier, and Emmanuel Malherbe. Learned hallucination detection in black-box LLMs using token-level entropy production rate. *arXiv preprint arXiv:2509.04492*, 2025. doi: 10.48550/arXiv.2509.04492. URL https://arxiv.org/abs/2509.04492.

Mengjia Niu, Hamed Haddadi, and Guansong Pang. Robust hallucination detection in LLMs via adaptive token selection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4213. URL https://arxiv.org/abs/2504.07863.

OpenAI. GPT-4.1 Mini. OpenAI API documentation. URL https://developers.openai.com/api/docs/models/gpt-4.1-mini. Accessed: September 21, 2026.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02707.

Ahmad Shapiro, Karan Taneja, and Ashok Goel. HALT: Hallucination assessment via log-probs as time series. *arXiv preprint arXiv:2602.02888*, 2026. doi: 10.48550/arXiv.2602.02888. URL https://arxiv.org/abs/2602.02888.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2024. doi: 10.1145/3637528.3671796. URL https://doi.org/10.1145/3637528.3671796.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. jina-embeddings-v3: Multilingual embeddings with task LoRA, 2024. URL https://arxiv.org/abs/2409.10173.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. CoCoA: A minimum bayes risk framework bridging confidence and consistency for uncertainty quantification in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/98d2006bf15f00e2ff293dfba4acb8b2-Paper-Conference.pdf.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL https://arxiv.org/abs/2212.03533.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL https://arxiv.org/abs/2002.10957.

Huanyi Xie, Yubin Chen, Liangyu Wang, Lijie Hu, and Di Wang. Predicting LLM output length via entropy-guided representations. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.11812.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

A AI USE STATEMENT

In preparing this manuscript, we used LLMs primarily for language polishing and, to a lesser extent, for identifying potentially relevant literature; we verified all cited references ourselves. DeepSeek-V4-Flash was separately used as an LLM-based judge to assign correctness labels for AUROC evaluation, with a random subset manually inspected. We retain full responsibility for all intellectual contributions and have reviewed and verified the final manuscript and all AI-assisted outputs.

B ADDITIONAL METHOD DETAILS

B.1 LOCAL CONSISTENCY PROPAGATION

Learnable Compatibility Term. The compatibility score $\eta_{t,k}$ in Eq. (6) captures a learnable semantic interaction between the Top-1 token and the k-th alternative candidate, complementing the explicit consistency cues used for local gating. It is computed as

$$\eta_{t,k} = q_{\text{pair}}(\text{LN}([\mathbf{y}_t; \mathbf{c}_{t,k}])) \,, \tag{16}$$

where LN denotes layer normalization and q_{pair} is a two-layer MLP with a SiLU activation between the two linear layers and a scalar output.

Iterative Local Propagation. We perform Q_L rounds of local propagation, keeping the graph structure and candidate representations fixed while updating only the center node representation. Across propagation rounds, the semantic similarity, rank distance, log-probability gap, and Top-K entropy remain fixed, while the learnable compatibility score is recomputed using the updated center representation. We then update the center representation according to Eq. (1), using the output of the previous round as the center input. Each round uses a separate set of learnable parameters. The final center representation is denoted by $\mathbf{y}_t^{\text{L}}$ and passed to the global anticipatory consistency module.

B.2 LONGER-RANGE CONSISTENCY PROPAGATION

Probability Computation under Top-K Observations. The relative anticipatory probability in Eq. (8) is computed using only the available Top-K records. For each candidate $c \in C_j$, we use its recorded probability $p_i(c)$ if it appears among the Top-K candidates at step i; otherwise, its contribution to the denominator is set to zero.

Iterative Global Propagation. We perform Q_G layers of global propagation, keeping the graph structure fixed while updating the answer node representations. Across propagation layers, the anticipatory probability, relative anticipatory probability, semantic relatedness, and relative semantic relatedness remain fixed. Starting from $\{\mathbf{y}_t^{\text{L}}\}_{t=1}^{L}$, we successively apply Eq. (12), using the output of each layer as the node representations for the next layer. The final node representations are denoted by $\{\mathbf{y}_t^{\text{G}}\}_{t=1}^{L}$ and passed to the dual-anchor readout module.

B.3 SEMANTIC AND TEMPORAL ANCHOR READOUT

Semantic Anchor Initialization. Before training, we collect the unique answer tokens from the training set and apply frequency-weighted spherical k-means to their initial node representations, using token occurrence frequencies as weights. We initialize the semantic anchors with the resulting cluster centers. During training, these anchors are updated jointly with the remaining detector parameters.

Dual-Anchor Fusion Details. The projection ψ in Eq. (15) is implemented as

$$\psi(\mathbf{u}_m) = \text{Dropout}(\text{GELU}(\mathbf{W}_\psi \, \text{LN}(\mathbf{u}_m) + \mathbf{b}_\psi)) \,, \tag{17}$$

where $\mathbf{W}_\psi$ projects the input to 64 dimensions. The temporal projection ψ' has the same architecture with independent parameters.

The weighting function f applies a linear score to the projected features and normalizes the scores across semantic anchors:

$$f(\mathbf{u}_m) = \frac{\exp\Big(\mathbf{w}_f^\top \psi(\mathbf{u}_m) + b_f\Big)}{\sum_{m'=1}^{M} \exp\Big(\mathbf{w}_f^\top \psi(\mathbf{u}_{m'}) + b_f\Big)}, \tag{18}$$

where $\mathbf{w}_f$ and b_f are learnable parameters. The function g follows the same formulation with independent scoring parameters, using ψ' and normalizing over the N temporal anchors.

C Experimental Details

C.1 Data Splits and Label Construction

Data Splits. We construct answer-level hallucination detection datasets from TriviaQA, HaluEval, HotpotQA, and GSM8K. Each example consists of a question, a reference answer, an answer generated by the target model, and the corresponding per-step Top-K candidate tokens and probabilities.

For open-source target models, we perform a single teacher-forced forward pass on each generated answer to extract the Top-20 candidate tokens and their log-probabilities at each generation position. All data splits are fixed before detector training. The sample counts used in the main experiments are reported in Table 7.

Table 7: Data splits for answer-level hallucination detection tasks.

Dataset	Train	Valid	Test
TriviaQA	8,500	1,500	10,000
HaluEval	6,800	1,200	2,000
HotpotQA	8,500	1,500	10,000
GSM8K	6,352	1,121	1,319

Label Construction. We use a uniform LLM-as-a-judge protocol across all datasets to construct binary correctness labels. Specifically, we provide DeepSeek-V4-Flash with the question, the reference answer supplied by the dataset, and the target model's generated answer, and prompt the judge to determine whether the generated answer correctly addresses the question. For examples with multiple reference answers or answer aliases, all references are provided simultaneously to the judge.

We use the following prompt template for the judge across all datasets:

Prompt for Answer Correctness Labeling

System:
You are an answer correctness evaluator. Determine whether a model-generated answer correctly answers the question with respect to the reference answer(s).
Judge semantic correctness rather than exact string matching. Accept equivalent paraphrases, aliases, abbreviations, numerical forms, and harmless formatting differences.
Return only: `{"label": "correct"}` or `{"label": "incorrect"}`
Do not provide any explanation.

User:
Question: `{question}`
Reference answer(s): `{reference answers}`
Model answer: `{model answer}`

Figure 3: Prompt used for automatic answer correctness labeling.

If the judge's initial output cannot be parsed, we resubmit the same input and request a properly formatted response. We use the resulting binary labels to train and evaluate the detector.

Manual Verification. We manually verify a random subset of 300 responses, including 100 generated by DeepSeek-V4-Flash and 200 generated by the other target LLMs. We report the DeepSeek subset separately because DeepSeek serves as both a target LLM and the annotation judge. Human judgments agree with the automatic labels on 91% of DeepSeek responses and 93% of responses from the other models, yielding an overall agreement rate of 92.33%.

C.2 Prompt Templates

Below, we present the prompt templates used to generate answers from the target models.

Open-Domain Question Answering. We use a concise question-answer format for TriviaQA, HaluEval, and HotpotQA:

Prompt for Open-Domain Question Answering

User:
Q: {question}
A:

Figure 4: Prompt used for answer generation on TriviaQA, HaluEval, and HotpotQA.

Mathematical Reasoning. For GSM8K, we prompt the model to reason step by step and provide the final numerical answer in a fixed format, so as to evaluate whether Prop-K remains effective on chain-of-thought responses:

Prompt for GSM8K Mathematical Reasoning

User:
Question: {question}

Please reason step by step. On the final line, output only:

Final answer: `\boxed{<number>}`

Do not use `\boxed{}` for intermediate results.

Figure 5: Prompt used for answer generation on GSM8K.

C.3 Fixed Implementation Settings

External Semantic Encoder. We use a frozen `all-MiniLM-L6-v2` model as the external semantic encoder $\mathbb{E}$. We independently encode each decoded token with the external encoder, apply ℓ_2 normalization to the resulting 384-dimensional representations, and cache them before training. The node mapping ϕ projects the external representations into the detector's node representation space and is trained jointly with the detector.

External Embedding Models. For the embedding ablation in Table 4, we additionally evaluate E5 (Wang et al., 2024) (`intfloat/e5-large-v2`), Jina (Sturua et al., 2024) (`jinaai/jina-embeddings-v3`), and GTE (Li et al., 2023b) (`thenlper/gte-base`). All encoders remain frozen and follow the same preprocessing procedure described above.

Hyperparameter Tuning. We keep all other experimental settings fixed and tune only the hyperparameters listed in Table 8. Specifically, we select the number of local propagation rounds Q_{L} from $\{1, 2\}$, the number of global propagation rounds Q_{G} from $\{1, 2\}$, and the global window size W

from $\{8, 12\}$. All models are trained with constant learning rates without warmup, using binary cross-entropy loss. The learning rate for the anchor module is fixed at 0.1 times the base learning rate, and the random seeds are set to $\{42, 43, 44\}$.

We use the same search space for all target language model–dataset combinations. For each combination, we first select the checkpoint with the highest validation AUROC for each configuration, then select the hyperparameter configuration with the highest validation AUROC. The test set is used only for final evaluation.

Table 8: Hyperparameter search space. The same search grid is used for all model–dataset combinations.

Hyperparameter	Candidate values
Learning rate	$\{10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$
Dropout	$\{0.1, 0.2, 0.3\}$
Number of local propagation rounds Q_{L}	$\{1, 2\}$
Number of global propagation rounds Q_{G}	$\{1, 2\}$
Global window size W	$\{8, 12\}$

C.4 Implementation Details of Baseline Methods

Black-Box Baselines. For black-box methods that require multiple samples, we generate 20 candidate answers per question and share the same set across these methods. Both SelfCheckGPT-NLI and Semantic Entropy use `potsawee/deberta-v3-large-mnli` for natural language inference, with the question and answer provided together as input and a maximum sequence length of 512 tokens. For Semantic Entropy, we use the discrete variant based on sampling frequencies. For LLM-Judge, we use GPT-4o as the judge and prompt it to output a `yes`/`no` judgment based solely on the question and the answer being evaluated. Reference answers are not provided to this baseline. We report a single run for all black-box baselines due to the high computational and token costs of repeating response sampling or API-based judging across multiple seeds.

Gray-Box Baselines. All gray-box baselines use the Top-20 candidate tokens and their log-probabilities from a single generation record as input features. We implement LOS-Net using its publicly available code[1] and adapt it to our Top-K record format. We reimplement EPR and WEPR (Moslonka et al., 2025), as well as HALT (Shapiro et al., 2026), following the descriptions in their original papers. We evaluate all gray-box baselines except EPR over three random seeds. EPR is evaluated once because its scores are deterministic given fixed generation records.

White-Box Baselines. For all white-box baselines, we follow the feature extraction procedures described in the original papers, while using a common protocol for hidden-layer selection. Specifically, SAPLMA (Azaria & Mitchell, 2023) uses the hidden state of the last valid answer token as input to the detector. HaMI (Niu et al., 2025) uses the hidden states of all answer tokens[2]. CHARM (Frasca et al., 2026) constructs node features from the hidden states of both prompt and answer tokens, builds the graph using the model's internal attention, and performs readout only at answer positions[3]. We evaluate all white-box baselines over three random seeds.

For these three methods, we predefine four candidate layers at 40%, 60%, 80%, and 100% of the model's total depth, rounding each resulting layer index up to the nearest integer and excluding the input embedding layer from the depth count. Each baseline selects one of these layers based on validation AUROC and keeps the selected layer fixed for test-set evaluation. The test set is not used for hidden-layer selection.

[1] `https://github.com/BarSGuy/Beyond-next-token-probabilities`

[2] HaMI code: `https://github.com/mala-lab/HaMI`

[3] CHARM code: `https://github.com/Noired/charm`

D SUPPLEMENTARY EXPERIMENTS

D.1 ROBUSTNESS ACROSS RANDOM SEEDS

The main text reports mean test AUROC over three random seeds (42, 43, and 44); here, we additionally provide the corresponding sample standard deviations.

Main Results. The following tables report test AUROC as mean ± sample standard deviation across three random seeds.

Table 9: Test AUROC (↑) across four datasets and four open-source target LLMs (three seeds).

Mistral-7B

Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
WEPR	0.656±0.000	0.668±0.000	0.670±0.000	0.659±0.000	0.663
LOS-Net	0.676±0.018	0.691±0.017	0.664±0.007	0.657±0.006	0.672
HALT	0.779±0.004	0.751±0.008	0.762±0.014	0.750±0.043	0.761
White-box methods					
SAPLMA	0.802±0.001	0.753±0.010	0.742±0.003	0.786±0.001	0.771
HaMI	**0.875**±0.003	0.788±0.005	**0.818**±0.005	0.829±0.001	0.828
CHARM	0.864±0.001	**0.811**±0.007	0.812±0.002	0.843±0.000	**0.833**
Prop-K	0.846±0.003	0.808±0.006	0.808±0.001	**0.855**±0.010	0.829

Llama3-8B

Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
WEPR	0.732±0.000	0.735±0.000	0.711±0.000	0.713±0.000	0.723
LOS-Net	0.723±0.002	0.670±0.005	0.656±0.003	0.721±0.008	0.693
HALT	0.788±0.004	0.720±0.019	0.748±0.004	0.816±0.045	0.768
White-box methods					
SAPLMA	0.780±0.008	0.828±0.002	0.794±0.004	0.786±0.007	0.797
HaMI	0.839±0.003	0.818±0.010	0.795±0.002	0.881±0.004	0.833
CHARM	0.858±0.004	**0.836**±0.005	**0.817**±0.004	0.887±0.005	**0.850**
Prop-K	**0.863**±0.006	0.824±0.005	0.801±0.007	**0.913**±0.011	**0.850**

Gemma3-4B

Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
WEPR	0.673±0.000	0.678±0.000	0.709±0.000	0.797±0.000	0.714
LOS-Net	0.740±0.002	0.744±0.009	0.755±0.002	0.700±0.027	0.735
HALT	0.810±0.002	0.802±0.023	0.806±0.013	0.842±0.005	0.815
White-box methods					
SAPLMA	0.822±0.001	0.731±0.033	0.752±0.015	0.817±0.056	0.781
HaMI	0.856±0.002	0.790±0.003	0.789±0.009	0.910±0.002	0.836
CHARM	**0.877**±0.000	0.836±0.005	0.820±0.004	0.890±0.003	0.856
Prop-K	0.846±0.002	**0.850**±0.003	**0.853**±0.008	**0.928**±0.003	**0.869**

Gemma3-12B

Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
WEPR	0.680±0.000	0.667±0.000	0.673±0.000	0.723±0.000	0.686
LOS-Net	0.733±0.001	0.713±0.003	0.717±0.002	0.646±0.019	0.702
HALT	0.794±0.003	0.766±0.037	0.782±0.014	0.809±0.012	0.788
White-box methods					
SAPLMA	0.852±0.002	0.809±0.014	0.795±0.008	0.610±0.003	0.767
HaMI	0.853±0.002	0.814±0.006	0.804±0.006	**0.879**±0.001	0.838
CHARM	**0.882**±0.002	**0.846**±0.002	**0.827**±0.000	0.846±0.004	**0.850**
Prop-K	0.872±0.001	**0.846**±0.007	0.822±0.003	0.848±0.014	0.847

As shown in Table 9, Prop-K consistently outperforms the gray-box baselines and performs comparably to the white-box baseline CHARM, surpassing it in several model–dataset settings. Prop-K also shows limited variation across random seeds.

Table 10: Test AUROC (↑) across four datasets and three closed-source target LLMs (three seeds).

GPT-4.1-mini					
Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
LOS-Net	0.722±0.006	0.754±0.003	0.713±0.000	0.734±0.012	0.731
HALT	0.848±0.001	0.821±0.005	0.796±0.001	0.735±0.018	0.800
Prop-K	**0.878**±0.006	**0.844**±0.004	**0.823**±0.004	**0.826**±0.014	**0.843**

Gemini-2.5-Flash-Lite					
Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
LOS-Net	0.794±0.001	0.854±0.001	0.743±0.002	0.838±0.005	0.807
HALT	0.854±0.002	0.856±0.002	0.820±0.003	0.853±0.014	0.846
Prop-K	**0.868**±0.003	**0.873**±0.007	**0.838**±0.007	**0.936**±0.001	**0.879**

DeepSeek-V4-Flash					
Method	TriviaQA	HaluEval	HotpotQA	GSM8K	Avg.
Gray-box methods					
LOS-Net	0.714±0.004	0.837±0.002	0.775±0.002	0.755±0.007	0.770
HALT	0.774±0.001	0.852±0.002	0.790±0.017	0.782±0.012	0.799
Prop-K	**0.852**±0.002	**0.874**±0.004	**0.833**±0.005	**0.836**±0.009	**0.849**

As shown in Table 10, Prop-K achieves a higher mean AUROC than both gray-box baselines in every closed-source model–dataset setting and shows limited variation across random seeds.

Ablation Results. Table 11 reports mean ± sample standard deviation across three random seeds for the four ablation studies in Section 4.2.

Table 11: Ablation results across three random seeds. Test AUROC (↑) is reported as mean ± sample standard deviation.

Category	Configuration	HaluEval	GSM8K
Propagation	w/o Local	0.791±0.007	0.798±0.004
	w/o Global	0.827±0.005	0.821±0.005
	w/o Both	0.714±0.006	0.726±0.008
	Prop-K	**0.846±0.007**	**0.848±0.014**
Global design	w/o Global	0.827±0.005	0.821±0.005
	Self-attention	0.830±0.010	0.834±0.018
	Prop-K ($W = 1$)	0.833±0.009	0.812±0.006
	Prop-K ($W \gg 1$)	**0.846±0.007**	**0.848±0.014**
Embedding	Jina	0.843±0.004	0.842±0.019
	E5	0.848±0.002	0.835±0.020
	GTE	0.843±0.008	0.857±0.013
	MiniLM (default)	**0.846±0.007**	**0.848±0.014**
Readout	Mean pooling	0.813±0.005	0.812±0.005
	Max pooling	0.831±0.013	0.832±0.006
	Attention pooling	0.824±0.007	0.822±0.015
	Dual-anchor	**0.846±0.007**	**0.848±0.014**

Prop-K benefits from full propagation and dual-anchor readout, with stable performance across embedding models and random seeds.

D.2 Statistical Significance

We compare Prop-K with each baseline using two-sided paired Student's t-tests over 16 model–dataset pairs, covering four open-source target LLMs and four datasets ($n = 16$). Each paired observation consists of the AUROC scores of Prop-K and the baseline evaluated on the same target LLM and dataset. We report unadjusted p-values.

Table 12: Two-sided paired t-test p-values comparing Prop-K with each baseline across 16 model–dataset pairs from four open-source target LLMs and four datasets.

Baseline	p
Black-box methods	
Self-consistency	5.32×10^{-9}
Lexical similarity	6.58×10^{-9}
LLM-Judge (GPT-4o)	6.25×10^{-4}
SelfCheckGPT-NLI	2.35×10^{-7}
Semantic Entropy	4.53×10^{-9}
Gray-box methods	
EPR	3.37×10^{-11}
WEPR	2.60×10^{-11}
LOS-Net	1.92×10^{-10}
HALT	1.12×10^{-8}
White-box methods	
SAPLMA	3.10×10^{-4}
HaMI	0.0369
CHARM	0.6874

As shown in Table 12, Prop-K achieves significant average AUROC gains over all evaluated black-box and gray-box baselines, as well as SAPLMA ($p < 0.01$). It also achieves a statistically significant gain over HaMI ($p = 0.0369$), whereas no statistically significant difference is detected relative to CHARM ($p = 0.6874$).

D.3 Analysis of Learned Propagation Weights

To examine the contribution of consistency-aware weighting beyond graph-based aggregation, we replace the learned gating weights $\alpha_{t,k}$ and/or $\beta_{i,j}$ with uniform weights in the local stage, the global stage, or both, while keeping the graph structure and all other components unchanged.

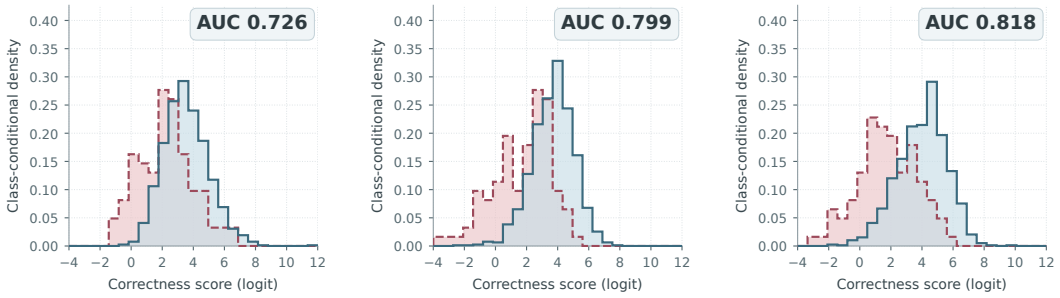
Table 13: Effect of learned propagation weights. Test AUROC (↑), reported as mean ± sample standard deviation over three seeds.

Configuration	HaluEval	GSM8K
Uniform weights (global)	$0.831_{\pm 0.001}$	$0.821_{\pm 0.016}$
Uniform weights (local)	$0.818_{\pm 0.002}$	$0.829_{\pm 0.023}$
Uniform weights (both)	$0.809_{\pm 0.002}$	$0.806_{\pm 0.006}$
Prop-K	$\mathbf{0.846}_{\pm 0.007}$	$\mathbf{0.848}_{\pm 0.014}$

As shown in Table 13, replacing the learned global or local propagation weights with uniform weighting reduces AUROC by about 0.02–0.03, while applying uniform weighting to both stages leads to larger drops of about 0.04. These results demonstrate that Prop-K benefits from selectively propagating consistency-relevant information rather than simply aggregating neighboring representations.

D.4 How does consistency propagation improve answer representations?

Using Gemma3-12B on GSM8K, we apply mean pooling to obtain answer-level representations before propagation, after LAC, and after GAC. We then train identically configured linear probes on these representations and visualize their test-set logit distributions. This analysis examines whether consistency propagation progressively enhances the discriminative information contained in external semantic representations by incorporating target-model-specific predictive signals.



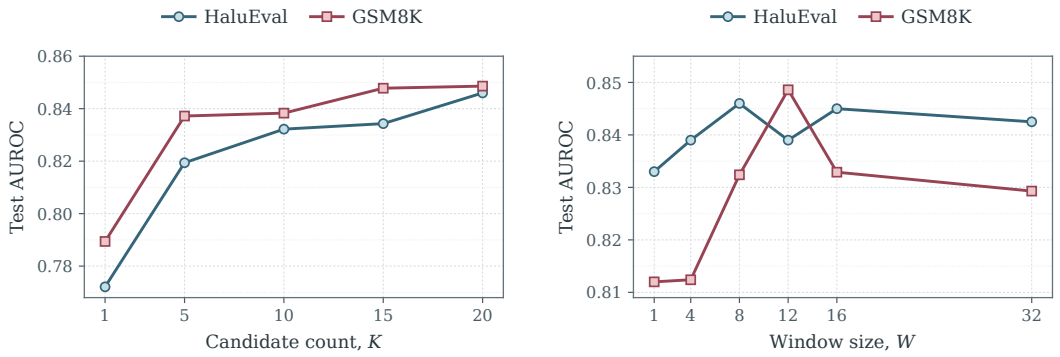
(a) Before propagation (b) After LAC (c) After GAC

Figure 6: Class-conditional distributions of correctness logits at different propagation stages. Blue and red denote correct and incorrect responses, respectively.

As shown in Figure 6, the overlap between correct and incorrect responses progressively decreases, with AUROC increasing from 0.726 to 0.799 and then to 0.818. This indicates that LAC and GAC progressively improve representation separability by incorporating local candidate agreement and global anticipatory consistency, respectively. These results suggest that consistency propagation enriches external semantic representations with target-model-dependent predictive patterns, leading to more discriminative answer-level representations for hallucination detection.

D.5 Sensitivity Analysis

We examine the sensitivity to the number of Top-K candidates and the anticipatory window size W on HaluEval and GSM8K using Gemma3-12B. We vary one parameter at a time while retaining the original experimental settings.



(a) Top-K candidate count. (b) Anticipatory window size W.

Figure 7: Sensitivity to the Top-K candidate count and anticipatory window size on HaluEval and GSM8K.

Number of Top-K Candidates. We vary $K \in \{1, 5, 10, 15, 20\}$ while keeping all other settings fixed. AUROC increases monotonically with K on both datasets, with the largest improvement

occurring between $K = 1$ and $K = 5$. Both datasets achieve their best performance at $K = 20$, supporting the use of multiple predictive candidates for consistency propagation.

Anticipatory Window Size. We vary $W \in \{1, 4, 8, 12, 16, 32\}$, changing only the maximum distance between connected answer positions. Performance initially improves as the window expands, peaking at $W = 8$ on HaluEval and $W = 12$ on GSM8K. For larger windows, HaluEval exhibits small fluctuations, whereas GSM8K shows a clearer performance degradation. These results suggest that the useful anticipatory range depends on the dataset and that a larger window does not necessarily improve detection.

D.6 Supplementary Experiments on Probability–Semantic Fusion

This section provides implementation details and results for the two fusion approaches introduced in Motivating Experiments. We evaluate both approaches on HaluEval and GSM8K using Gemma3-12B as the target language model and a frozen `all-MiniLM-L6-v2` encoder to obtain external semantic representations from token text.

Direct Feature Integration. Using HALT as the baseline, we examine the effects of adding external semantic features from Top-1 and Top-20 tokens. For Top-1, we encode the token actually generated at each generation step; for Top-20, we concatenate the external embeddings of the 20 candidate tokens at that step in probability rank order. We then project the resulting semantic features to 128 dimensions, concatenate them with HALT's original probability features, and feed the combined features into its subsequent temporal modeling and classification modules.

The original HALT achieves AUROC scores of 0.766 and 0.809 on HaluEval and GSM8K, respectively. Adding Top-1 external semantic features yields AUROC scores of 0.761 and 0.817, corresponding to a decrease of less than 0.01 and an increase of about 0.01, respectively. Adding Top-20 external semantic features yields AUROC scores of 0.774 and 0.818, corresponding to increases of about 0.01 on both datasets. In these experiments, Top-1 external semantic features do not consistently improve detection performance, while Top-20 external semantic features yield only small improvements, indicating that directly concatenating external semantic features provides limited gains.

Probability-Modulated Interaction. We further examine whether the probability modulation mechanism in the white-box method HaMI can be transferred to external semantic representations. Specifically, we replace the target model's internal hidden states used by HaMI with token representations from the external encoder, while retaining its probability modulation mechanism and subsequent detection architecture.

The white-box HaMI results obtained using the target model's internal hidden states are 0.814 and 0.879 in AUROC on HaluEval and GSM8K, respectively. Using external token representations reduces AUROC to 0.726 and 0.813, corresponding to absolute drops of approximately 0.09 and 0.07, respectively. This result indicates that the probability modulation mechanism used in the white-box setting does not readily maintain its detection performance when applied directly to external semantic representations.

E Availability of Top-K Decoding Metadata in Mainstream LLM APIs

Prop-K treats the target language model as a gray-box generator and does not require access to model parameters, hidden states, gradients, or logits over the full vocabulary. It requires the generation API to return, at each output position, the log-probability of the generated token, along with several high-probability candidate tokens and their log-probabilities. The API can return this metadata alongside the generated answer, eliminating the need to deploy the target model locally or perform additional forward passes.

We survey representative commercial model APIs as of August 31, 2026. Table 14 lists the corresponding models, access interfaces, and maximum numbers of returned candidates. Here, $K_{\max}$

denotes the maximum number of candidate tokens that the API allows to be returned at each generation position, rather than the vocabulary size considered by the model during decoding. Actual availability may still vary with the model, region, endpoint, and product version.

Table 14: Support for per-token Top-K log-probabilities in mainstream model APIs.

Model provider	Model	Access interface	$K_{\max}$
OpenAI	GPT	OpenAI API	20
Google	Gemini	Vertex AI	20
DeepSeek	DeepSeek	DeepSeek API	20
Moonshot AI	Kimi	Kimi API	20
Alibaba Cloud	Qwen	DashScope	5
Zhipu AI	GLM-4.7-Flash	Cloudflare Workers AI	20

The availability reported in the table is based on official documentation for the respective interfaces, including the OpenAI Responses API[4], Google Vertex AI generation parameters[5], the DeepSeek Chat Completions API[6], the Kimi Chat Completions API[7], the Alibaba Cloud DashScope API[8], and the GLM-4.7-Flash interface on Cloudflare Workers AI[9]. The `top_p` and `top_k` parameters only control the candidate set used for sampling during generation; they do not themselves provide candidate probabilities at each position. Prop-K requires the response to explicitly include log-probabilities for the generated token and its candidate tokens.

When an API returns fewer candidates than the $K = 20$ used in the main experiments, Prop-K can still construct the graph from the available candidates, but the observable candidate coverage is correspondingly reduced. Thus, the listed interfaces show that the per-token Top-K metadata required by Prop-K is publicly available through multiple independent model providers and access platforms, rather than being limited to a single model or a local deployment environment.

[4] https://platform.openai.com/docs/api-reference/responses/create

[5] https://cloud.google.com/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters

[6] https://api-docs.deepseek.com/api/create-chat-completion/

[7] https://platform.kimi.ai/docs/api/chat

[8] https://help.aliyun.com/en/model-studio/qwen-api-via-dashscope

[9] https://developers.cloudflare.com/workers-ai/models/glm-4.7-flash/