MAX-POOLING NETWORK REVISITED: ANALYZING THE ROLE OF SEMANTIC PROBABILITY IN MULTIPLE INSTANCE LEARNING FOR HALLUCINATION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hallucination detection is essential for reliable large language models (LLMs). Hybrid methods such as HaMI combine semantic consistency with internal states through multiple instance learning (MIL), but repeated sampling and semantic comparisons impose substantial computational overhead. We analyze HaMI's decision margins and identify conditions under which semantic-probability scaling enlarges the average logit margin. Motivated by this analysis, we revisit classical sentence classification from a local margin dynamics perspective: a shared MLP extracts token-level features, feature-wise max pooling aggregates them, and a linear head scores the response. The detector requires no additional sampling or semantic consistency computation. Across five benchmarks and three LLMs, it is competitive with semantically weighted HaMI in detection AUC while sustaining several thousand QA detections per second. Its larger gains over fixed-position probes on LongFact support aggregating evidence across token positions.

1 INTRODUCTION

Hallucinations hinder the reliable deployment of large language models (LLMs) (Huang et al., 2025; Ji et al., 2023; Longpre et al., 2021), motivating detectors that identify factual errors in generated responses. Existing approaches to hallucination detection fall into three main categories. First, uncertainty-based methods estimate confidence from model probabilities or from the semantic diversity of multiple sampled responses, as in Semantic Entropy (Farquhar et al., 2024), which accounts for semantic equivalence. Second, retrieval-based approaches query external knowledge sources or LLMs to verify factual consistency (Tan et al., 2025). Third, representation-based methods exploit internal hidden states (Azaria & Mitchell, 2023; Gao et al., 2025), often training classifiers on fixed token positions such as the final token (Orgad et al., 2025). A recent hybrid method, HaMI (Niu et al., 2025), formulates hallucination detection as a multiple instance learning (MIL) problem and improves robustness through adaptive token selection and semantic-probability weighting.

HaMI's semantic weighting requires repeated generation and external semantic comparisons, adding latency and API costs. Single-token hidden-state probes are fast, but commit to one token position and can miss evidence elsewhere in the response. We ask whether sequence-level detection can retain HaMI's accuracy while maintaining high detection throughput.

Despite its empirical success, it remains unclear why semantic-probability weighting improves detection performance. We analyze HaMI and show that semantic-probability scaling enlarges the average logit margin under an explicit condition on class proportions and scaling sensitivities. This motivates studying whether aggregation of internal token features can support margin expansion without semantic weighting.

Specifically, we revisit max-over-time sentence classification (Kim, 2014): a shared MLP transforms token states, feature-wise max pooling aggregates them, and a linear head scores the response. Different features can retain evidence from different tokens. We analyze each bag's margin increase under its own gradient update and show how max pooling avoids gradient dilution under a sparse-MIL assumption. This motivates aggregating informative features across token positions.

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.

Our contributions are summarized as follows:

- We characterize when semantic weighting enlarges HaMI's average margin (Theorem 1).
- We analyze margin dynamics under each bag's own update and sparse-signal aggregation in max-pooling networks (Theorems 2 and 3), and bound their capacity (Proposition 1).
- Across five benchmarks and three LLMs, our detector is competitive with HaMI (SP) on short-answer QA, performs strongly on LongFact at 256- and 1024-token limits, and sustains several thousand QA detections per second (Sections 6.2–6.4).

Table 1: Comparison of architectural designs between HaMI and our study.

Aspect	**HaMI Niu et al. (2025)**	**Ours**
Aggregation	Instance-based	Embedding-based
Pooling Method	TopK Pooling	Max Pooling Kim (2014)
Signals	Semantic Metric + Hidden States	Hidden States

2 Related Work

LLM Hallucination and Detection Metrics Factual errors can reflect failures to access internal knowledge (Zheng et al., 2023). Detection methods include logit-based indicators (Varshney et al., 2023) and consistency checks across sampled responses, such as SelfCheckGPT (Manakul et al., 2023).

Semantic Entropy (SE) quantifies uncertainty via semantic clustering (Farquhar et al., 2024); SE Probes approximate it from hidden representations to avoid repeated sampling and semantic comparisons (Kossen et al., 2024).

Internal States and Latent Truthfulness Recent probing studies suggest that truthfulness signals are not uniformly scattered throughout the model, but instead are concentrated in specific layers and token positions (Azaria & Mitchell, 2023; Orgad et al., 2025). At the level of individual neurons, prior work identified a small subset of hallucination-related neurons, termed H-Neurons, and showed that manipulating their activations can directly increase or decrease the model's tendency to generate false information (Gao et al., 2025).

Multiple Instance Learning (MIL) for Hallucination Detection Multiple Instance Learning (MIL) assigns labels to bags of instances (Carbonneau et al., 2018; Maron & Lozano-Pérez, 1997); attention-based deep MIL learns to weight their contributions (Ilse et al., 2018).

Building on this perspective, the HaMI framework (Niu et al., 2025) models hallucination as a local, token-dependent phenomenon rather than a uniform sentence-level property. It formulates detection as a MIL task, adaptively weighting token contributions using uncertainty metrics and internal representations, thereby outperforming fixed aggregation baselines.

In contrast, our work revisits this aggregation process (Kim, 2014) by relying solely on internal representations, eliminating the need for external uncertainty estimation while retaining the ability to capture informative token-level signals.

3 Preliminaries

3.1 Problem Settings

Hallucination refers to the phenomenon in which responses contradict factual reality, typically arising when LLMs lack sufficient internal knowledge or fail to properly use their internal knowledge (Zheng et al., 2023).

For short-answer QA, let $D = \{(q_i, a_i)\}_{i=1}^{N}$, where q_i is a question and a_i its reference answer. LongFact supplies prompts without reference answers. Given a question or prompt q, an LLM produces a response sequence $\mathcal{A} = (t_1, t_2, \ldots, t_n)$.

For each generated token t_i, we extract the hidden representation $\mathbf{h}_i \in \mathbb{R}^d$ from an intermediate layer. An external evaluator LLM assigns $y = +1$ to a hallucinated response and $y = -1$ to a factual response. This evaluator supplies supervision; our detector requires no external evaluator at inference.

3.2 Semantic Probability

Following the Semantic Entropy framework (Farquhar et al., 2024), we quantify uncertainty by analyzing the semantic diversity of stochastic samples $\mathcal{G} = \{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$ for each question q.

Semantic Clustering To group linguistically diverse but semantically equivalent responses, we use an external LLM to evaluate the relation $R_{ij} := R(\mathcal{A}_i, \mathcal{A}_j) \in \{E, C, N\}$ (*Entailment, Contradiction, Neutral*). We define equivalence as:

$$\mathcal{A}_i \equiv \mathcal{A}_j \iff (R_{ij}, R_{ji}) \in \{(E,E), (E,N), (N,E)\}$$

The sampled responses in $\mathcal{G}$ are grouped into semantic clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_s\}$.

Uncertainty Metrics Following HaMI (Niu et al., 2025), a cluster's probability is its empirical frequency among the K sampled responses:

$$P_{\mathcal{C}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}\{\mathcal{A}_k \in \mathcal{C}\}.$$

Each sampled response is counted separately, including repeated identical texts. Its **semantic probability** is:

$$P_{\text{sem}}^{(i)} = P_{\mathcal{C}(i)}, \quad \text{where } \mathcal{C}(i) \text{ is the cluster containing } \mathcal{A}_i.$$

3.3 Hallucination Detection via Multiple Instance Learning

We formulate hallucination detection as Multiple Instance Learning (MIL) (Carbonneau et al., 2018; Ilse et al., 2018; Maron & Lozano-Pérez, 1997). Each response forms a bag of token states, $\mathbf{B} = \{\mathbf{h}_{\mathbf{B},i}\}_{i=1}^{T_{\mathbf{B}}}$, with label $y_{\mathbf{B}}$. The dataset $\mathcal{S}$ contains the following positive and negative subsets:

$$\mathcal{S}_{pos} = \{\mathbf{B} \in \mathcal{S} \mid y_{\mathbf{B}} = 1\}, \quad \mathcal{S}_{neg} = \{\mathbf{B} \in \mathcal{S} \mid y_{\mathbf{B}} = -1\}$$

To ensure permutation invariance, the scoring function $S(\mathbf{B})$ must be symmetric.

Aggregation Strategies Two orders are possible. *Instance-based* aggregation scores each token first and pools the scores, $S(\mathbf{B}) = g_1(\{f_1(\mathbf{h}_{\mathbf{B},i})\}_{i=1}^{T_{\mathbf{B}}})$; *embedding-based* aggregation pools the hidden states into a bag embedding and scores once, $S(\mathbf{B}) = f_2(g_2(\{\mathbf{h}_{\mathbf{B},i}\}_{i=1}^{T_{\mathbf{B}}}))$. HaMI takes the first route, ours the second.

4 Analysis of HaMI (Niu et al., 2025)

4.1 HaMI Architecture Formulation

The HaMI architecture is illustrated in Figure 1. For a bag $\mathbf{B} = \{\mathbf{h}_{\mathbf{B},i}\}_{i=1}^{T_{\mathbf{B}}}$ with $\mathbf{h}_{\mathbf{B},i} \in \mathbb{R}^d$, a network f turns each instance into a logit, and HaMI scales the instances by the semantic probability $P_{\text{sem}}^{\mathbf{B}}$ before doing so:

$$z_{\mathbf{B},i} = f(\mathbf{h}_{\mathbf{B},i}) = w^\top \text{ReLU}(\text{BN}(\mathbf{x}_{\mathbf{B},i} + \mathbf{b}_1)) + b_2, \qquad \tilde{\mathbf{h}}_{\mathbf{B},i} = (1 + \lambda P_{\text{sem}}^{\mathbf{B}})\mathbf{h}_{\mathbf{B},i},$$

where $\mathbf{x}_{\mathbf{B},i} = \mathbf{W}\mathbf{h}_{\mathbf{B},i}$, BN is batch normalisation, $\lambda > 0$, and $\tilde{z}_{\mathbf{B},i} = f(\tilde{\mathbf{h}}_{\mathbf{B},i})$. The bag score $S(\mathbf{B})$ averages the TopK instance scores $\mathcal{K}_{\mathbf{B}}$.

Margin Formulation For a dataset $\mathcal{S}$ of bags, define the average logit-space margin by

$$M_{\text{hami}}(\lambda) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{B} \in \mathcal{S}} y_{\mathbf{B}} \tilde{Z}_{\mathbf{B}}, \qquad \tilde{Z}_{\mathbf{B}} = \frac{1}{k} \sum_{i \in \mathcal{K}_{\mathbf{B}}} \tilde{z}_{\mathbf{B},i}.$$

We use logit space because the sigmoid is monotone but not distance-preserving: at large $|z|$, it compresses margin differences into negligible probability differences.

4.2 Margin Enhancement via Asymmetric Scaling

We analyze the effect of the scaling factor $p_{\mathbf{B}} = \lambda P^{\mathbf{B}}_{\text{sem}}$. Because batch normalisation is affine at inference and the ReLU active set $\mathcal{A}_{\mathbf{B},i}$ is locally invariant under the scaling, the scaled logit separates exactly into an unscaled part and a term linear in $p_{\mathbf{B}}$ (Appendix B.1):

$$\tilde{z}_{\mathbf{B},i} = z_{\mathbf{B},i} - p_{\mathbf{B}} C(\mathbf{x}_{\mathbf{B},i}), \qquad C(\mathbf{x}_{\mathbf{B},i}) = -\frac{\partial \tilde{z}_{\mathbf{B},i}}{\partial p_{\mathbf{B}}} = -\sum_{j \in \mathcal{A}_{\mathbf{B},i}} w_j \gamma_j \frac{x_{\mathbf{B},i,j}}{\sigma_j}.$$

For clarity, we present the derivation assuming a locally invariant active set. However, using the property that ReLU networks are continuous piecewise-affine (CPWA) maps (Arora et al., 2018), we can generalize this sensitivity analysis via path integration. This generalization ensures that our formulation $Z_B(p_B) = Z_B(0) - p_B \bar{C}_B^{\text{int}}$ remains mathematically rigorous even under dynamic changes in activation patterns, as derived in Appendix C.

Empirically, we observe the following three properties (Appendix G):

$$\bar{C}_{\mathbf{B}} > 0, \quad \text{where} \quad \bar{C}_{\mathbf{B}} = \frac{1}{k} \sum_{i \in \mathcal{K}_{\mathbf{B}}} C(\mathbf{x}_{\mathbf{B},i}) \tag{1}$$

$$\mathbb{E}_{neg}[P^{\mathbf{B}}_{\text{sem}}] > \mathbb{E}_{pos}[P^{\mathbf{B}}_{\text{sem}}] \tag{2}$$

$$\mathbb{E}_{neg}[\bar{C}_{\mathbf{B}}] > \mathbb{E}_{pos}[\bar{C}_{\mathbf{B}}]. \tag{3}$$

These properties can be understood from the following perspectives:

- **Logit Suppression Effect:** The positivity in Equation (1) indicates that the scaling operation inherently acts as a penalty. The scaling operation suppresses the overall logits.
- **Semantic Consistency:** Equation (2) reflects the intuition that models produce more semantically consistent outputs for non-hallucinated instances.
- **Input-Scale Sensitivity:** Driven by the classification loss during training, the model adaptively adjusts $\bar{C}_{\mathbf{B}}$. It learns to increase $\bar{C}_{\mathbf{B}}$ for negative instances to amplify logit suppression, while decreasing it for positive instances to prevent undesired attenuation. This adaptive behavior naturally leads to Equation (3).

Motivated by these observations, we define the ratio of the expected product of the scaling factor and logit sensitivity between the negative and positive classes as γ:

$$\gamma = \frac{\mathbb{E}_{neg}[P^{\mathbf{B}}_{\text{sem}} \bar{C}_{\mathbf{B}}]}{\mathbb{E}_{pos}[P^{\mathbf{B}}_{\text{sem}} \bar{C}_{\mathbf{B}}]}. \tag{4}$$

The empirical validation of this property and the observed range of γ are provided in Appendix G.1.

Theorem 1 (Margin Enhancement Logic via Semantic Probability Weighting)**.** *Weighting the input instances by their semantic probabilities strictly increases the empirical expected margin* $\mathbb{E}_{\mathbf{B}}[M_{hami}]$*, provided that the dataset class ratio satisfies the following condition:*

$$\frac{|\mathcal{S}_{neg}|}{|\mathcal{S}_{pos}|} > \frac{1}{\gamma}. \tag{5}$$

Interpretation. Theorem 1 formally justifies the mechanism of HaMI. Since the scaling operation inherently suppresses logits, it introduces a trade-off: it expands margins for negative bags while compressing them for positive ones. The parameter γ characterizes the relative magnitude of this effect, representing the ratio of expected margin gain in negative bags to the penalty in positive bags. Consequently, Equation 5 implies that as long as the dataset contains a sufficient proportion of negative bags to leverage this sensitivity differential, the overall expected margin will increase. This widened margin establishes a more robust decision boundary for hallucination detection.

5 MAX POOLING NETWORK REVISITED (KIM, 2014)

HaMI adjusts token-level scales using semantic information. However, since it scores tokens independently, it may not fully capture sentence-level coherence. We therefore revisit the classical sentence classification model (Kim, 2014) and study how aggregation affects a bag's margin increase.

5.1 MODEL ARCHITECTURE

The model's architecture is shown in Figure 2. It treats each LLM token's hidden state as an instance and computes the hallucination probability for the entire bag.

Let a bag be $\mathbf{B} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{T_\mathbf{B}}\}$ with $\mathbf{h}_i \in \mathbb{R}^d$ the hidden state of the i-th generated token. A shared projection maps each instance into a D-dimensional feature space, the bag representation takes the maximum along the token axis independently per dimension, and a linear layer produces the score:

$$\mathbf{u}_i = \mathrm{ReLU}(W^\top \mathbf{h}_i), \qquad v_j = \max_{i \le T_\mathbf{B}} (u_i)_j, \qquad z_\mathbf{B} = w^\top \mathbf{v}, \qquad S(\mathbf{B}) = \sigma(z_\mathbf{B}).$$

This is the max-over-time pooling of Kim (2014), not a maximum over instance scores as in TopK MIL: the bag representation may combine evidence from different tokens in different dimensions. With $y_\mathbf{B} = +1$ for hallucinated bags and -1 for factual bags, training minimises the logistic loss $\mathcal{L}_\mathbf{B} = \log(1 + \exp(-y_\mathbf{B} z_\mathbf{B}(\theta)))$. Max pooling could be applied to the raw d-dimensional hidden states directly. Section 5.3 bounds the capacities of both operation orders, and Section 6.6 compares their detection accuracy.

5.2 MARGIN ANALYSIS

Bag-wise Margin. Our analysis and Section 6.5 use the signed bag logit as the *bag-wise margin*:

$$m_\mathbf{B}(\theta) := y_\mathbf{B}\, z_\mathbf{B}(\theta), \tag{6}$$

Positive margins indicate correct classification, and their magnitude is the logit-space distance from zero. The logistic loss $\mathcal{L}_\mathbf{B} = \log(1 + \exp(-m_\mathbf{B}(\theta)))$ decreases with the margin. This matches M_{hami} for the TopK-pooled logit $\tilde{Z}_\mathbf{B}$ (Section 4.1); $\mathcal{M}(\theta) = \mathbb{E}_\mathbf{B}[m_\mathbf{B}(\theta)]$ is the expected bag-wise margin.

Theorem 2 (Expected Margin Expansion under Per-Bag Updates)**.** *For a finite empirical bag distribution, define each bag's own update $\theta_{\mathbf{B},\eta} := \theta - \eta \nabla_\theta \mathcal{L}_\mathbf{B}(\theta)$ and $\mathcal{M}_\eta(\theta) := \mathbb{E}_\mathbf{B}[m_\mathbf{B}(\theta_{\mathbf{B},\eta})]$, so $\mathcal{M}_0(\theta) = \mathcal{M}(\theta)$. For sufficiently small $\eta > 0$ with each update staying in a smooth region,*

$$\mathcal{M}_\eta(\theta) = \mathcal{M}_0(\theta) + \eta \mathbb{E}_\mathbf{B}\left[\frac{\|\nabla_\theta z_\mathbf{B}(\theta)\|^2}{1 + \exp(m_\mathbf{B}(\theta))}\right] + O(\eta^2).$$

The increase is strict when the expected first-order coefficient is positive.

Each bag is evaluated after its own update; this does not assert an increase in the mean margin of a shared updated model. Appendix B.7 gives an explicit step-size condition under a β-smoothness assumption.

Motivation: Hallucination Detection as Sparse MIL. We model hallucination detection as a sparse MIL problem, motivated by the observation that factual errors may be localized to a subset of a response (Maynez et al., 2020; Niu et al., 2025). In this setting, mean pooling can dilute sparse informative features by averaging them with background features, whereas max pooling preserves the largest activation in each dimension and routes gradients through the corresponding tokens. Motivated by prior analyses of max and mean pooling (Boureau et al., 2010), we examine this distinction under the following assumption.

Assumption 1 (Sparse MIL Structure)**.** *For fixed $\theta = (W, w)$, let $\mathbf{g}_{\mathbf{B},i,j} := \mathbf{h}_{\mathbf{B},i} \mathbf{1}\{W_j^\top \mathbf{h}_{\mathbf{B},i} > 0\}$. For each bag $\mathbf{B}$, let $S_{\mathbf{B},j} \subseteq \{1, \ldots, T_\mathbf{B}\}$ be the set of informative instances ($s_{\mathbf{B},j} := |S_{\mathbf{B},j}|$), $s_\mathbf{B} := \max_j s_{\mathbf{B},j}$, and $J_\mathbf{B} := \{j : s_{\mathbf{B},j} \ge 1\}$. We assume:*

1. ***Sparsity:*** *$i \notin S_{\mathbf{B},j} \implies u_{\mathbf{B},i,j} = 0$ and $\mathbf{g}_{\mathbf{B},i,j} = \mathbf{0}$.*

2. ***Boundedness:*** *For all $j \in J_{\mathbf{B}}$ and $i \in S_{\mathbf{B},j}$, there exist positive constants such that $u_{\mathbf{B},i,j} \in [\underline{u}, \overline{u}]$ and $\|\mathbf{g}_{\mathbf{B},i,j}\|_2 \in [\underline{g}, \overline{g}]$.*

Theorem 3 (Gradient Norm Ratio in Sparse MIL)**.** *Under Assumption 1 for any positive bag $\mathbf{B}$, the ratio of the gradient norms for max pooling and mean pooling satisfies:*

$$\frac{\|\nabla_\theta z_{\mathbf{B}}^{\max}\|^2}{\|\nabla_\theta z_{\mathbf{B}}^{\text{mean}}\|^2} = \Omega\left(\left(\frac{T_{\mathbf{B}}}{s_{\mathbf{B}}}\right)^2\right).$$

Gradient Dynamics under Per-Bag Updates. Theorem 2 links each bag's margin gain under its own update to its squared logit-gradient norm and the factor $(1+\exp(m_{\mathbf{B}}))^{-1}$. Theorem 3 compares the gradient norms at fixed parameters: max pooling avoids their dilution by averaging in sparse positive bags. This supports larger first-order gains when the margin-dependent factors are matched; the gradient-norm comparison alone does not order gains at different margins or establish faster training of a shared model. Although the result is restricted to positive bags under Assumption 1, it provides a theoretical explanation for how max pooling can preserve sparse informative signals that would otherwise be diluted by averaging.

5.3 Rademacher Complexity Analysis

We bound the Rademacher complexity (Bartlett & Mendelson, 2002) of both pooling orders under shared norm constraints.

Setup. Let $\mathcal{S} = \{\mathbf{B}_i\}_{i=1}^n$ be a set of bags, where $\mathbf{B}_i = \{h_{i,t}\}_{t=1}^{T_i} \subset \mathbb{R}^d$, $\|h_{i,t}\|_2 \le R$, and $T = \max_i T_i$. We compare two max-pooling architectures: the model $\mathcal{F}_{\text{base}}$ that pools raw input features directly, and the model $\mathcal{F}_{\text{feat}}$ that transforms instances into a latent space before pooling.

With ReLU activation $a(x) = \max\{0, x\}$, instance-wise max pooling ρ, and weight bounds $\|W_j\|_2 \le B_1$ and $\|w\|_2 \le B_2$, their hypothesis classes are defined as:

$$\mathcal{F}_{\text{feat}} = \left\{w^\top \rho\big(\{a(W_j h_t)\}_{j=1,\dots,D;\, t=1,\dots,T}\big)\right\},$$
$$\mathcal{F}_{\text{base}} = \left\{w^\top a\big(W\,\rho(\mathbf{B})\big)\right\}.$$

Proposition 1 (Rademacher Complexity Bounds)**.** *Under the setup above, the empirical Rademacher complexities are bounded as follows:*

$$\hat{\mathcal{R}}_S(\mathcal{F}_{\text{feat}}) \;\le\; \frac{2\sqrt{2}\,RB_1B_2\sqrt{DT}}{\sqrt{n}}, \quad \hat{\mathcal{R}}_S(\mathcal{F}_{\text{base}}) \;\le\; \frac{2RB_1B_2\sqrt{Dd}}{\sqrt{n}}.$$

Discussion. These norm-based upper bounds characterize the capacity of each architecture, but their comparison alone does not establish an ordering of the actual complexities or a generalization advantage for either architecture (Appendix B.6). The benefit of transforming features before pooling is supported empirically by the placement ablation in Section 6.6.

6 Experiments

6.1 Experimental Setup

Models and data. We evaluate LLaMA-3.1-8B (base), Mistral-Nemo-12B (Instruct), and LLaMA-3.3-70B (Instruct, 4-bit quantized) (Dubey et al., 2024; Mistral AI, 2024) on TriviaQA (Joshi et al., 2017), SQuAD (Rajpurkar et al., 2016), NQ (Kwiatkowski et al., 2019) and BioASQ (Krithara et al., 2023), with a 100-token generation limit. Mean test-response lengths are 6–35 tokens; LongFact (Wei et al., 2024) is evaluated separately (Section 6.3). We generate five responses per question in every split at temperature 0.5, using only one for detector training and all five for validation and test. Before filtering, train/validation/test counts are 4,000/4,500/5,500, except BioASQ (2,300/3,000/3,000; Appendix A.1).

Baselines. We compare Semantic Entropy Probes (SEP) (Kossen et al., 2024), the last-token probe of Orgad et al. (2025) (LLMs Know), SAPLMA (Azaria & Mitchell, 2023), HaMI and HaMI (SP) (Niu et al., 2025), and mean and max pooling. Methods share generations, factuality labels, splits and hardware; hidden-state extraction is also shared.

Table 2: Mean QA AUC over five runs. TQ: TriviaQA; SQ: SQuAD; BQ: BioASQ. Red/blue: best/second-best.

	LLaMA-3.1-8B				Mistral-Nemo-12B				LLaMA-3.3-70B			
Method	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ
SEP	0.821	0.759	0.729	0.792	0.842	0.773	0.730	0.836	0.870	0.824	0.829	0.855
LLMs Know	0.879	0.779	0.766	0.872	0.902	0.810	0.802	0.882	0.934	0.863	0.871	0.888
SAPLMA	0.870	0.776	0.765	0.861	0.905	0.837	0.834	0.880	0.909	0.866	0.884	0.900
HaMI	0.865	0.795	0.777	0.862	0.915	0.840	0.837	0.889	0.907	0.851	0.871	0.874
HaMI (SP)	0.923	0.840	0.840	0.903	0.932	0.869	0.859	0.894	0.926	0.856	0.878	0.880
Mean	0.897	0.815	0.789	0.885	0.924	0.854	0.846	0.902	0.911	0.857	0.870	0.877
Max	0.924	0.827	0.815	0.899	0.936	0.871	0.867	0.913	0.937	0.880	0.897	0.884

Labels and training. GPT-5 mini (OpenAI, 2025b) judges QA against reference answers and LongFact from internal knowledge, as it has no references. Initially hallucinated responses are re-checked using internal knowledge and discarded only if judged factual. Agreement with human TruthfulQA labels (Lin et al., 2022) is 95.3% (Appendix D.3). We train for 100 epochs with batch size 128, select checkpoints and layers by validation AUC, and report five-run means. Hidden-state methods share the stride-two layer search of Niu et al. (2025); mean and max pooling also share $D = 256$ and tuning budgets. Prompts, implementation and hyperparameters are in Appendix A.1; selected layers and standard errors are in Appendices A.4 and E.3.

6.2 Classification Performance

Table 2 tests whether aggregating token features improves response-level detection. Max pooling outperforms mean pooling in all twelve model–dataset pairs, with average gains of 1.5–2.1 AUC points across the four QA benchmarks. The two methods share the feature-extraction architecture, feature dimension and tuning budget, making the pooling rule the key difference in this comparison. A feature-wise maximum can retain a strong activation even when most tokens do not express that feature. These gains are consistent with preserving token-level cues that averaging may attenuate.

Max pooling also outperforms unweighted HaMI in all twelve settings. Its QA averages are 0.866, 0.897 and 0.900, versus 0.877, 0.889 and 0.885 for HaMI (SP). Thus, aggregating learned features from a single response can be competitive with semantic weighting that requires additional responses. Semantic weighting remains strongest on LLaMA-3.1-8B, while max pooling leads on the larger models.

Max pooling improves the four-benchmark average over SAPLMA by 1.0–4.8 points and over LLMs Know by 1.1–4.8 points. SAPLMA remains best on LLaMA-3.3-70B/BioASQ (0.900 versus 0.884), so gains are not universal. Pooling gives the classifier direct access to features from multiple token positions.

6.3 Detection on Long Responses

We evaluate LongFact at 256- and 1024-token generation limits with corresponding length instructions in the prompt (Appendix D.2). Sampling and judging follow Section 6.1. Mean lengths across validation and test rise from 188 to 391 tokens (8B), 237 to 922 (12B), and 241 to 983 (70B). Before filtering, detector train/validation/test counts are 1,000/1,000/1,500. Each method's validation-selected layer at 256 tokens is fixed across both settings.

HaMI, HaMI (SP), mean pooling and max pooling all obtain higher mean AUC than both fixed-position probes in every LongFact condition (Table 3). Although a fixed-position hidden state can encode earlier context, MIL lets the detector read token states throughout the answer directly. The advantage across these different aggregation rules supports treating a long response as a bag of token states, with useful detection cues potentially occurring at multiple positions. This formulation also learns from a label for the entire response, without requiring token-level error annotations.

Table 3: Mean LongFact AUC over five runs at two generation limits (tokens). Colors follow Table 2.

	LLaMA-3.1-8B		Mistral-Nemo-12B		LLaMA-3.3-70B	
Method	256 tokens	1024 tokens	256 tokens	1024 tokens	256 tokens	1024 tokens
SEP	0.567	0.584	0.517	0.499	0.528	0.484
LLMs Know	0.648	0.748	0.619	0.669	0.655	0.720
SAPLMA	0.624	0.742	0.608	0.631	0.625	0.707
HaMI	0.683	0.818	0.672	0.716	0.730	0.783
HaMI (SP)	0.675	0.815	0.673	0.714	0.722	0.784
Mean	0.768	0.851	0.669	0.728	0.753	0.742
Max	0.781	0.861	0.750	0.849	0.781	0.788

Table 4: Mean QA throughput (samples/sec). Red/blue: best/second-best.

Method	LLaMA 3.1-8B	Mistral 12B	LLaMA 3.3-70B
SEP	7,711	7,955	7,406
LLMs Know	7,526	7,401	7,102
SAPLMA	4,753	4,748	4,558
HaMI (Original)	3,802	3,095	2,327
HaMI (SP)	0.13	0.19	0.15
Mean Pooling	5,085	4,167	2,586
Max Pooling	5,273	4,181	2,564

Table 5: Mean QA test margins. Red/blue: higher/lower within each block.

Method	LLaMA 3.1-8B	Mistral 12B	LLaMA 3.3-70B
HaMI (Original)	3.93	4.35	4.10
HaMI (SP)	6.80	5.86	5.66
Mean Pooling	1.28	1.71	1.89
Max Pooling	1.48	1.99	2.14

Max pooling achieves the highest mean AUC in all six conditions. Each learned feature can select a different token, allowing the bag representation to combine evidence from different parts of a response. This may help retain useful cues distributed across long answers. At 1024 tokens, max pooling exceeds the fixed-position probes by 6.8–21.8 AUC points. Its gap over mean pooling on Mistral-Nemo grows from 8.1 points at 256 tokens to 12.1 at 1024. On 70B/1024, however, HaMI (0.783) and HaMI (SP) (0.784) are close to max pooling (0.788), suggesting that instance-score aggregation can also be competitive in this setting.

The two generation settings also differ in their prompts and label distributions, so the AUC changes cannot be attributed to length alone. The 12B and 70B Instruct models have only 69 and 35 hallucinated training responses at 1024 tokens; differences in instruction following may contribute to this imbalance (Appendix D.2).

6.4 Efficiency Comparison

On one A100 without batching, max pooling scores extracted hidden states at 2,564–5,273 samples/sec, averaged over the four QA benchmarks (Table 4). HaMI (SP) achieves 0.09–0.28 samples/sec across datasets, including its additional sampling and semantic clustering. Both measurements exclude the initial response generation. Thus, sequence-wide aggregation retains throughput comparable in magnitude to lightweight probes, while avoiding the sampling cost of semantic weighting. Per-dataset timings are in Appendix E.1.

6.5 Margin Results

Table 5 reports mean test margins $m_{\mathbf{B}} = y_{\mathbf{B}} z_{\mathbf{B}}$ after 100 epochs at validation-selected layers; HaMI uses its TopK-pooled logit. HaMI (SP) raises the QA mean by 35–73% over HaMI, and max pooling by 13–16% over mean pooling (full results in Appendix E.2). These directions agree with Sections 4.2 and 5.2. However, margins depend on logit scale, so we compare them descriptively

Table 6: Feature-placement ablation (mean AUC, five shared seeds; same validation-selected layer). Abbreviations and colors follow Table 2.

	LLaMA-3.1-8B				Mistral-Nemo-12B				LLaMA-3.3-70B			
Pooling	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ
Raw (d): pool the hidden states, then apply the MLP												
Mean	0.898	0.815	0.790	0.885	0.925	0.855	0.847	0.903	0.913	0.857	0.870	0.876
Max	0.891	0.807	0.782	0.872	0.929	0.856	0.849	0.896	0.880	0.848	0.861	0.868
Features (D): extract features, then pool the D-dimensional projections												
Mean	0.897	0.815	0.789	0.885	0.924	0.854	0.846	0.902	0.911	0.857	0.870	0.877
Max	0.924	0.827	0.815	0.899	0.936	0.871	0.867	0.913	0.937	0.880	0.897	0.884

within architecture families. The per-bag update analysis does not guarantee margins after shared training, and larger unnormalized margins alone do not imply higher AUC.

6.6 Architectural Ablations

To examine how token features should be aggregated, we keep the MLP architecture fixed and apply it either before or after pooling. Transforming tokens first improves max-pooling AUC by 0.7–5.7 points, while mean pooling changes by at most 0.2 point (Table 6). The operation order also changes the relative performance of the pooling rules: max pooling falls below mean pooling in nine of twelve settings when applied to raw states, but leads in all twelve after feature extraction. On LLaMA-3.3-70B/TriviaQA, max pooling rises from 0.880 to 0.937, while mean pooling changes from 0.913 to 0.911.

This reversal suggests that max pooling benefits from selecting learned token features. The token-wise MLP can form useful patterns before pooling combines coordinates across tokens. Pooling raw states first may lose within-token relationships before the MLP can use them. The results therefore support learning token-level features before aggregation. This is empirical evidence for the operation order; the capacity bounds alone do not establish an accuracy ordering.

Increasing D improves on a single-feature representation, with diminishing returns around $D = 128$–256; a matched mean/max sweep uses the same dimension grid (Appendix F.2). Layer-wise curves show performance across representation depths (Appendix F.3).

7 Conclusion

We revisited feature-wise max pooling for hallucination detection. Our analysis characterizes when semantic weighting enlarges HaMI's margin and relates each bag's margin increase under its own gradient update to sparse-MIL aggregation. The capacity analysis makes the role of feature dimension explicit, and the placement ablation shows the benefit of transforming hidden states before pooling. Together these motivate preserving task-relevant token features until aggregation, letting the classifier combine evidence from different positions in the response.

Across five benchmarks and three LLMs, max pooling is competitive with HaMI (SP) on short-answer QA and leads the tested detectors on LongFact at both 256- and 1024-token limits. It sustains several thousand QA detections per second without extra sampling, providing a baseline for comparing detection accuracy and cost. The LongFact results suggest that MIL can also be useful beyond short-answer QA. Evaluation with independent human annotations would help determine whether the observed gains also hold under human judgments.

Future work should evaluate responses beyond the current 1024-token budget while controlling for prompt content and class balance, to better understand how response length affects detection performance. A further direction is to move beyond response-level classification and investigate whether token-level features can help localise factual errors within individual claims and identify invalid steps in mathematical reasoning.

AI USE STATEMENT

We used LLaMA-3.1-8B and LLaMA-3.3-70B (Dubey et al., 2024) and Mistral-Nemo-12B (Mistral AI, 2024) to generate the responses evaluated in this study. GPT-5 mini (OpenAI, 2025b) provided factuality judgments, and GPT-4o mini (OpenAI, 2024) provided semantic entailment judgments for response clustering. The generation, labeling, and clustering procedures are described in Section 6.1 and Appendix A.1; validation of the automated labels against human annotations is reported in Appendix D.3.

We also used generative AI assistance through OpenAI Codex (OpenAI, 2025a) and Claude Code (Anthropic, 2025) to improve the manuscript's readability and organization, edit LaTeX and tables, check numerical summaries against reported results, review mathematical statements, assumptions, and proofs, and refine the interpretation of experimental results and their limitations. We further used these tools to make the experimental and appendix sections more concise and to generate tables from experimental data. The resulting tables and reported numerical values were checked by the authors against the underlying experimental data. The authors take responsibility for the final manuscript, including the correctness and originality of its claims, proofs, results, and any content produced with AI assistance.

REFERENCES

Anthropic. Claude 3.7 Sonnet and Claude Code, February 2025. URL https://www.anthropic.com/news/claude-3-7-sonnet.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units, 2018. URL https://arxiv.org/abs/1611.01491.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying, 2023. URL https://arxiv.org/abs/2304.13734.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, November 2002. URL https://jmlr.org/papers/v3/bartlett02a.html.

Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning*, ICML'10, pp. 111–118, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077. URL https://icml.cc/Conferences/2010/papers/638.pdf.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, May 2018. ISSN 0031-3203. doi: 10.1016/j.patcog.2017.10.009. URL https://doi.org/10.1016/j.patcog.2017.10.009.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783v1.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://www.nature.com/articles/s41586-024-07421-0.

Cheng Gao, Huimin Chen, Chaojun Xiao, Zhiyi Chen, Zhiyuan Liu, and Maosong Sun. H-Neurons: On the existence, impact, and origin of hallucination-associated neurons in LLMs, 2025. URL https://arxiv.org/abs/2512.01797.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55, January 2025. ISSN 1046-8188. doi: 10.1145/3703155. URL https://doi.org/10.1145/3703155.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning, 2018. URL https://arxiv.org/abs/1802.04712.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL https://arxiv.org/abs/1705.03551.

Yoon Kim. Convolutional neural networks for sentence classification, 2014. URL https://arxiv.org/abs/1408.5882.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs, 2024. URL https://arxiv.org/abs/2406.15927.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific Data*, 10:170, 2023. doi: 10.1038/s41597-023-02068-4. URL https://doi.org/10.1038/s41597-023-02068-4.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565/.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models, 2023. URL https://arxiv.org/abs/2303.08896.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In M. Jordan, M. Kearns, and S. Solla (eds.), *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL https://proceedings.neurips.cc/paper_files/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf.

Andreas Maurer. A vector-contraction inequality for rademacher complexities, 2016. URL https://arxiv.org/abs/1605.00251.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization, 2020. URL https://arxiv.org/abs/2005.00661.

Mistral AI. Mistral NeMo, July 2024. URL https://mistral.ai/news/mistral-nemo/.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks, 2014. URL https://arxiv.org/abs/1402.1869.

Mengjia Niu, Hamed Haddadi, and Guansong Pang. Robust hallucination detection in LLMs via adaptive token selection, 2025. URL https://arxiv.org/abs/2504.07863.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence, July 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

OpenAI. Introducing Codex, May 2025a. URL https://openai.com/index/introducing-codex/.

OpenAI. Introducing GPT-5 for developers, August 2025b. URL https://openai.com/index/introducing-gpt-5-for-developers/.

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, 2025. URL https://arxiv.org/abs/2410.02707.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. Too consistent to detect: A study of self-consistent errors in LLMs, 2025. URL https://arxiv.org/abs/2505.17656.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation, 2023. URL https://arxiv.org/abs/2307.03987.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. Long-form factuality in large language models, 2024. URL https://arxiv.org/abs/2403.18802.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. Why does ChatGPT fall short in providing truthful answers?, 2023. URL https://arxiv.org/abs/2304.10513.

The appendix provides implementation details (Appendix A), proofs and extensions (Appendices B and C), data and label validation (Appendix D), detailed detection results (Appendix E), ablations (Appendix F), and HaMI sensitivity checks (Appendices G and H).

A Implementation Details

A.1 Implementation and Training

Architecture and software. HaMI follows Niu et al. (2025): a two-layer MLP projects each hidden state to 256 dimensions, applies batch normalization and ReLU, and produces an instance logit. Mean and max pooling apply the same type of token-wise transformation, aggregate the transformed features, and use a linear output head. We use PyTorch $2.5.1$ and Transformers $5.5.3$.

Optimization and selection. Mean and max pooling use $D = 256$ and Adam with learning rate 2×10^{-4} and weight decay 5×10^{-3}. HaMI uses Adam with learning rate 10^{-3}, weight decay 5×10^{-4}, and TopK aggregation over 10% of tokens; HaMI (SP) uses $\lambda = 1$. Detectors train for 100 epochs at batch size 128 on one NVIDIA A100, retaining the checkpoint with the highest validation AUROC. Mean and max pooling share the dimension grid $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$ (Appendix F.2) and the stride-two layer search (Appendix A.4), with the same search budget and remaining hyperparameters.

Training takes approximately 140, 155, and 220 seconds for LLaMA-3.1-8B, Mistral-Nemo-12B, and LLaMA-3.3-70B, respectively, excluding data generation and labeling. Throughput is measured on one A100, processing each sample independently without batching.

Questions and generated responses. We generate five responses per question in every split, without an additional context passage. Detector training uses one response per question; validation and test use all five before filtering. The training splits contain $4{,}000$ questions each for TriviaQA, SQuAD, and NQ, $2{,}300$ for BioASQ, and $1{,}000$ for LongFact. Five responses per question yield $20{,}000$, $11{,}500$, and $5{,}000$ training-split generations, respectively, but only one response per question is used as a detector-training example, giving $4{,}000$, $2{,}300$, and $1{,}000$ examples before filtering. Validation contains 900, 600, and 200 questions, respectively, yielding $4{,}500$, $3{,}000$, and $1{,}000$ responses. Test contains $1{,}100$, 600, and 300 questions, yielding $5{,}500$, $3{,}000$, and $1{,}500$ responses. Thus, the combined validation and test totals are $10{,}000$, $6{,}000$, and $2{,}500$ generated responses. Tables 10 and 12 report retained responses after filtering for QA and LongFact, respectively; these counts differ from question counts and total generations.

Prompt 1: Short-answer QA.
The four QA benchmarks share the following template from Farquhar et al. (2024), as in Niu et al. (2025), with generation capped at 100 new tokens and stopped at the first blank line. LongFact uses Prompt 2.

```
Answer the following question in a single but complete sentence only.
Question: {question}
Answer:
```

Prompt 2: LongFact generation.
LongFact ships its own prompts, which we use verbatim with a length instruction appended; no single-sentence instruction, no `Question:`/`Answer:` framing and no passage are added. Newlines are removed from the stop sequences so that a multi-sentence answer is not cut at the first line break, and an answer that reaches the limit is trimmed back to its last complete sentence where one exists. The requested word count and number of aspects vary with the generation limit (Appendix D.2); a base model, which continues text rather than following the instruction, additionally receives the start of the enumeration.

```
{longfact_prompt}

Please provide a detailed answer of about {W} words that covers {K}
    distinct
aspects, numbered 1. to {K}., each explained in four to five sentences.
```

Labeling and semantic clustering. GPT-5 mini compares QA responses with reference answers. LongFact has no references, so both judgments use internal knowledge without retrieval. The re-check below is applied only to initially hallucinated responses; those judged factual on re-check are discarded, and all others retain their initial labels (Appendix D.3).

Prompt 3: Knowledge-only labeling re-check.

```
We are assessing the quality of answers to the following question:
    {question}
The proposed answer is: {predicted_answer}
Based on the context of question and your own knowledge, is the proposed
    answer correct?
Please think carefully and respond only with yes or no.
Response:
```

We employ GPT-4o mini as a natural language inference (NLI) judge to assess the semantic relationship between responses.

Prompt 4: Entailment.

```
Context: Question asked to an AI: '{question}'
Answer A: {text1}
Answer B: {text2}
Determine if Answer A entails Answer B.
Respond with only one word: entailment,contradiction, or neutral.
```

A.2 HaMI Architecture

The architecture of HaMI is illustrated in Figure 1. The model operates on an instance-based MIL framework, where each instance in a bag is first scaled by its semantic probability P_{sem}. These scaled representations are then processed through a two-layer MLP f to compute individual instance logits. To determine the bag-level score, HaMI employs a TopK pooling strategy, which selects and averages the most discriminative instance scores.

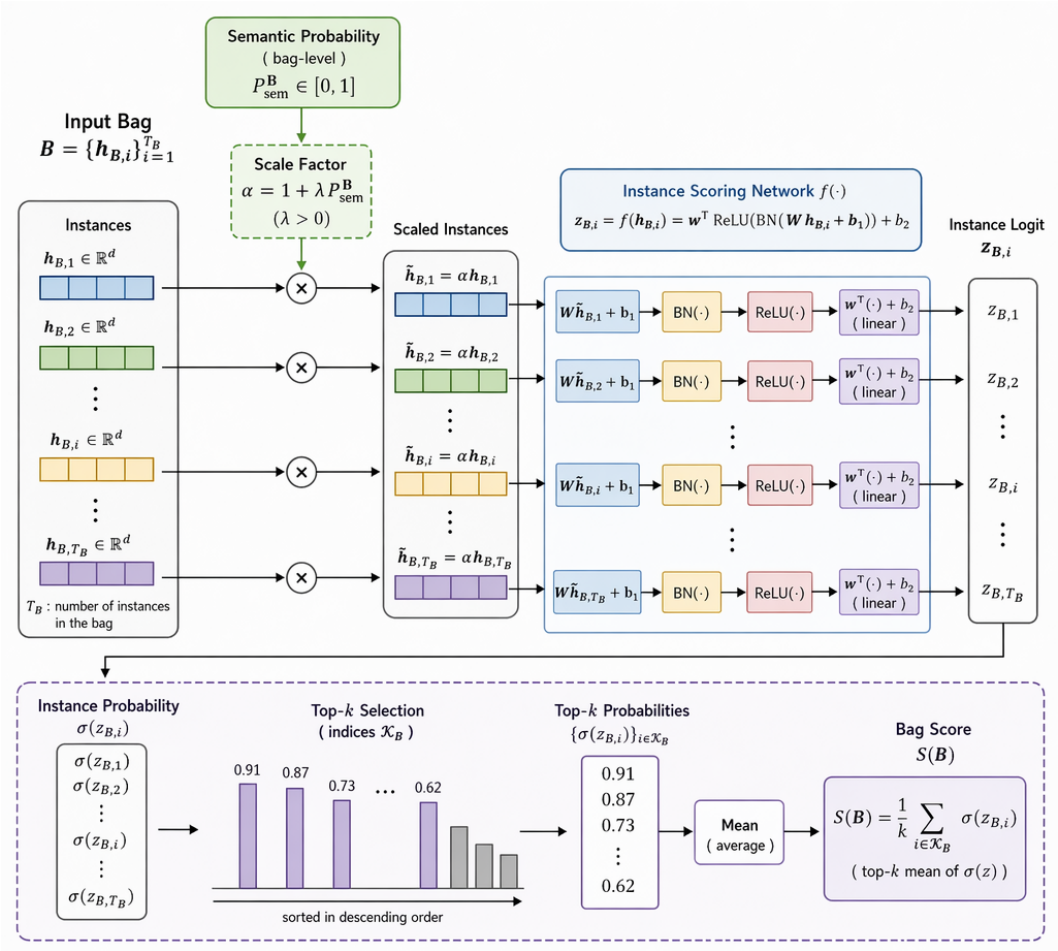


Figure 1: HaMI architecture.

A.3 Max Pooling Architecture

Figure 2 illustrates the response-level detector. Each state is projected into a lower-dimensional space, and max pooling is used to extract a bag representation by selecting the most salient features. A linear classifier then produces a hallucination probability.

The model is trained with a binary classification objective. Section 5.2 analyzes its local margin dynamics under sparse-MIL assumptions.

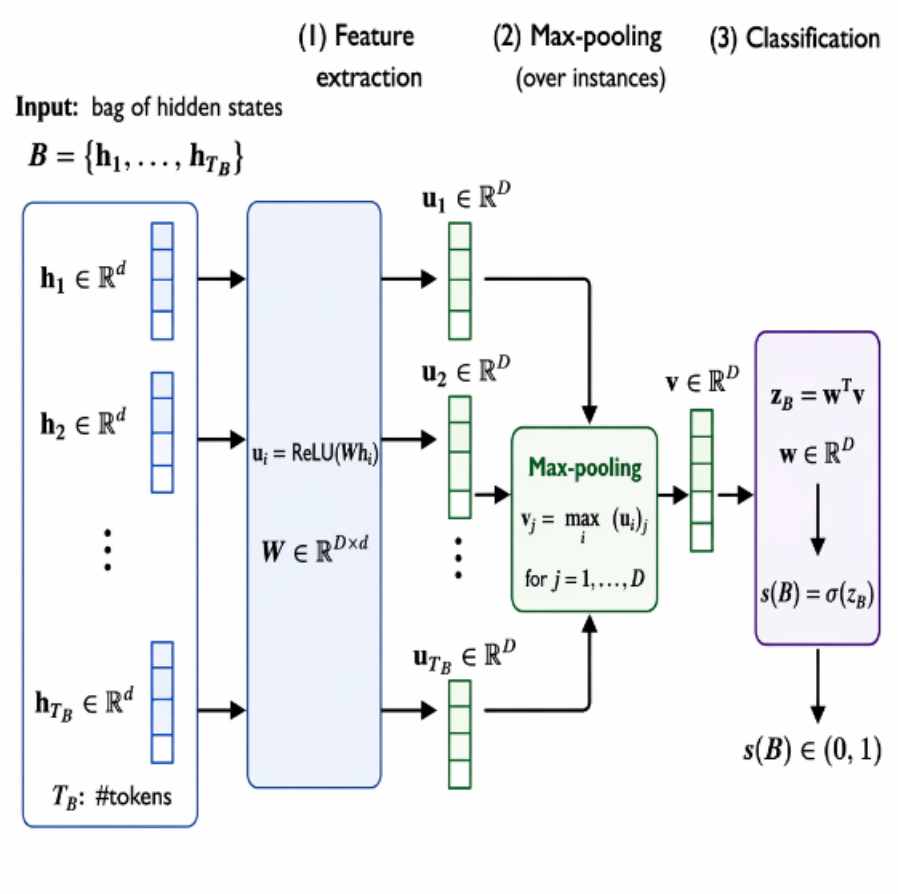


Figure 2: Max-pooling architecture. The projection matrix drawn as W corresponds to $W^\top$ in Section 5.1.

A.4 Selected Layers

The main-text results are reported at the layer that achieves the best validation AUC for each method. To make that choice auditable, we state the protocol and list the resulting layer indices.

Transformer layers were searched with a stride of two (layers $0, 2, 4, \ldots$), a detector was trained independently at each searched layer, and the layer with the highest validation AUC was retained. **The identical protocol, search stride, and search range were applied to our method and to every hidden-state baseline**, so no method received a wider layer search than any other. Selection used the validation split only; the test split was never consulted during selection.

Tables 7–9 list the selected indices. On the four short-answer benchmarks the selection concentrates near the middle of the network: the median selected layer is 14 of 32 for LLaMA-3.1-8B, 18 of 40 for Mistral-Nemo-12B and 36 of 80 for LLaMA-3.3-70B, i.e. 44–45% of depth in every case, consistent with prior probing results (Azaria & Mitchell, 2023; Orgad et al., 2025). The spread around that median is small for the two smaller generators (layers 12–16 and 16–20), so competing methods are read out at essentially the same depth and the reported differences reflect the aggregation strategy rather than the layer choice. LLaMA-3.3-70B is less uniform: most methods select layers 30–46, but on BioASQ the probe and sampling baselines move to 50–54 and HaMI (SP) to 76, and mean pooling selects 76 on TriviaQA. Those are the cells where the validation curve is flattest near its peak, so the arg-max is weakly determined. LongFact layers are selected separately for the response-length study (Table 14). Appendix F.3 shows the full layer-wise curves behind these choices.

Table 7: Selected transformer layers for LLaMA-3.1-8B (32 layers).

Method	TriviaQA	SQuAD	NQ	BioASQ
SEP	16	14	14	14
LLMs Know (last token)	14	14	14	14
SAPLMA	14	14	14	14
HaMI (Original)	12	14	14	14
HaMI (SP)	12	12	12	14
Mean Pooling	14	14	14	14
Max Pooling	14	14	14	14

Table 8: Selected transformer layers for Mistral-Nemo-Instruct (12B) (40 layers).

Method	TriviaQA	SQuAD	NQ	BioASQ
SEP	16	18	18	18
LLMs Know (last token)	16	18	18	18
SAPLMA	16	18	20	18
HaMI (Original)	16	18	18	18
HaMI (SP)	16	16	18	18
Mean Pooling	16	18	18	18
Max Pooling	16	18	18	18

Table 9: Selected transformer layers for LLaMA-3.3-Instruct-70B (80 layers).

Method	TriviaQA	SQuAD	NQ	BioASQ
SEP	42	36	34	50
LLMs Know (last token)	36	38	36	50
SAPLMA	46	36	36	54
HaMI (Original)	38	34	34	36
HaMI (SP)	36	34	32	76
Mean Pooling	76	34	32	32
Max Pooling	32	34	34	30

B Proofs and Additional Theory

B.1 Batch-normalisation decomposition

The step condensed in Section 4.2 in full.

We analyze the effect of the scaling factor $p_{\mathbf{B}} = \lambda P_{\text{sem}}^{\mathbf{B}}$. The pre-normalization feature with scaling during the forward pass is $(1 + p_{\mathbf{B}})\mathbf{x}_{\mathbf{B},i} + \mathbf{b}_1$. During inference, the running mean μ_j and standard deviation σ_j are fixed. The j-th dimension of the BN output is decomposed as:

$$\begin{aligned}\text{BN}((1 + p_{\mathbf{B}})\mathbf{x}_{\mathbf{B},i} + \mathbf{b}_1)_j &= \gamma_j \frac{(1 + p_{\mathbf{B}})x_{\mathbf{B},i,j} + b_{1,j} - \mu_j}{\sigma_j} + \beta_j \\ &= \text{BN}(\mathbf{x}_{\mathbf{B},i} + \mathbf{b}_1)_j + p_{\mathbf{B}}\gamma_j \frac{x_{\mathbf{B},i,j}}{\sigma_j}.\end{aligned}$$

Let $\mathcal{A}_{\mathbf{B},i}$ denote the active set of indices where the ReLU activation is strictly positive. Assuming the active set remains invariant under the scaling $p_{\mathbf{B}}$, the exact scaled logit $\tilde{z}_{\mathbf{B},i}$ becomes:

$$\tilde{z}_{\mathbf{B},i} = \sum_{j \in \mathcal{A}_{\mathbf{B},i}} w_j \left(\text{BN}(\mathbf{x}_{\mathbf{B},i} + \mathbf{b}_1)_j + p_{\mathbf{B}}\gamma_j \frac{x_{\mathbf{B},i,j}}{\sigma_j} \right) + b_2 = z_{\mathbf{B},i} - p_{\mathbf{B}} C(\mathbf{x}_{\mathbf{B},i}),$$

where $C(\mathbf{x}_{\mathbf{B},i}) = -\frac{\partial \tilde{z}_{\mathbf{B},i}}{\partial p_{\mathbf{B}}} = -\sum_{j \in \mathcal{A}_{\mathbf{B},i}} w_j \gamma_j \frac{x_{\mathbf{B},i,j}}{\sigma_j}$.

B.2 Proof of Theorem 1

Fix the network and the local ReLU and TopK selections. The same derivation applies to the integrated sensitivity in Appendix C. Assume both class-conditional product expectations are positive, as in the theorem. Given the exact decomposition $\tilde{z}_{\mathbf{B},i} = z_{\mathbf{B},i} - \lambda P_{\text{sem}}^{\mathbf{B}} C(\mathbf{x}_{\mathbf{B},i})$, the change in the bag-level logit, $\Delta Z_{\mathbf{B}} = \tilde{Z}_{\mathbf{B}} - Z_{\mathbf{B}}$, is exactly:

$$\Delta Z_{\mathbf{B}} = -\lambda P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}}.$$

The total margin change $\Delta M_{hami} = M_{hami}(\lambda) - M_{hami}(0)$ is the sum of contributions from the positive and negative bags:

$$\begin{aligned}\Delta M_{hami} &= \frac{1}{|\mathcal{S}|} \left(\sum_{\mathbf{B} \in \mathcal{S}_{pos}} (+1)\Delta Z_{\mathbf{B}} + \sum_{\mathbf{B} \in \mathcal{S}_{neg}} (-1)\Delta Z_{\mathbf{B}} \right) \\ &= \frac{\lambda}{|\mathcal{S}|} \left(\sum_{\mathbf{B} \in \mathcal{S}_{neg}} P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}} - \sum_{\mathbf{B} \in \mathcal{S}_{pos}} P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}} \right).\end{aligned}$$

By expressing the sums as empirical averages over the respective subsets, the overall expected margin change over the dataset $\mathcal{S}$ is exactly:

$$\begin{aligned}&\Delta M_{hami} \\ &= \lambda \left(\frac{|\mathcal{S}_{neg}|}{|\mathcal{S}|} \mathbb{E}_{\mathbf{B} \in \mathcal{S}_{neg}}[P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}}] - \frac{|\mathcal{S}_{pos}|}{|\mathcal{S}|} \mathbb{E}_{\mathbf{B} \in \mathcal{S}_{pos}}[P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}}] \right) \\ &= \lambda \mathbb{E}_{\mathbf{B} \in \mathcal{S}_{pos}}[P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}}] \frac{|\mathcal{S}_{pos}|}{|\mathcal{S}|} \left(\gamma \frac{|\mathcal{S}_{neg}|}{|\mathcal{S}_{pos}|} - 1 \right),\end{aligned}$$

where $\mathbb{E}_{\mathbf{B} \in \mathcal{S}_{neg}}$ and $\mathbb{E}_{\mathbf{B} \in \mathcal{S}_{pos}}$ denote the empirical expectations. From this expression, it follows that the expected margin increases, if and only if:

$$\frac{|\mathcal{S}_{neg}|}{|\mathcal{S}_{pos}|} > \frac{1}{\gamma}.$$

Thus, asymmetric scaling selectively deepens the negative logits on average, increasing the expected logit-space margin.

B.3 Proof of Bag-level Margin Dynamics under Logistic Loss

Lemma 4 (Bag-level Margin Dynamics). *For a sampled bag* $\mathbf{B}$ *at parameters in a smooth region, its own logistic-loss update produces a positive first-order change in its signed margin* $m_B = y_{\mathbf{B}} z_B$*, whenever* $\nabla_\theta z_B \neq 0$*. The magnitude of this increase is determined by the squared gradient norm scaled by the current margin's confidence.*

This lemma characterizes the local fitness of the model to the bag B. It shows that the logistic loss naturally induces a gradient flow that pushes the model's parameters in a direction that enlarges the margin of the current sample. Importantly, it establishes that the magnitude of this margin increase is directly proportional to the squared norm of the gradient.

Proof. We first derive the first-order change in the bag-level margin. Let $\mathcal{L}_{\mathbf{B}}(\theta) = \log(1 + \exp(-y_{\mathbf{B}} z_{\mathbf{B}}(\theta)))$ be the logistic loss for a bag B. Its gradient with respect to $\theta = \{W, w\}$ is:

$$\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta) = \frac{\partial \mathcal{L}_{\mathbf{B}}}{\partial z_{\mathbf{B}}} \nabla_\theta z_{\mathbf{B}}(\theta) = -\frac{y_{\mathbf{B}}}{1 + \exp(y_{\mathbf{B}} z_{\mathbf{B}}(\theta))} \nabla_\theta z_{\mathbf{B}}(\theta).$$

For a fixed starting parameter θ_t, the update induced by bag $\mathbf{B}$ is $\theta_{\mathbf{B},\eta} = \theta_t - \eta \nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta_t)$. Its parameter shift is:

$$\theta_{\mathbf{B},\eta} - \theta_t = \eta \frac{y_{\mathbf{B}}}{1 + \exp(m_{\mathbf{B}}(\theta_t))} \nabla_\theta z_{\mathbf{B}}(\theta_t),$$

where $m_{\mathbf{B}}(\theta) = y_{\mathbf{B}} z_{\mathbf{B}}(\theta)$. For a sufficiently small learning rate η, the first-order Taylor expansion of $z_{\mathbf{B}}(\theta_{\mathbf{B},\eta})$ around θ_t yields:

$$\begin{aligned} z_{\mathbf{B}}(\theta_{\mathbf{B},\eta}) &= z_{\mathbf{B}}(\theta_t) + \nabla_\theta z_{\mathbf{B}}(\theta_t)^\top (\theta_{\mathbf{B},\eta} - \theta_t) + O(\eta^2) \\ &= z_{\mathbf{B}}(\theta_t) + \eta \frac{y_{\mathbf{B}}}{1 + \exp(m_{\mathbf{B}}(\theta_t))} \|\nabla_\theta z_{\mathbf{B}}(\theta_t)\|^2 + O(\eta^2). \end{aligned}$$

Multiplying both sides by $y_{\mathbf{B}}$ and using $y_{\mathbf{B}}^2 = 1$, we obtain the margin increase:

$$\Delta m_{\mathbf{B}}(\theta_t) = \eta \frac{\|\nabla_\theta z_{\mathbf{B}}(\theta_t)\|^2}{1 + \exp(m_{\mathbf{B}}(\theta_t))} + O(\eta^2).$$

For a nonzero gradient and a sufficiently small step remaining in a smooth region, the positive first-order term dominates the remainder.

The squared gradient norm $\|\nabla_\theta z_{\mathbf{B}}\|^2$ can be decomposed into contributions from the classification weights w and feature extraction weights W. Since θ consists of $\{w_j, W_j\}_{j=1}^D$, we have:

$$\|\nabla_\theta z_{\mathbf{B}}\|^2 = \sum_{j=1}^{D} \left(\left| \frac{\partial z_{\mathbf{B}}}{\partial w_j} \right|^2 + \left\| \nabla_{W_j} z_{\mathbf{B}} \right\|_2^2 \right) = \sum_{j=1}^{D} \gamma_{B,j}.$$

For **Max Pooling**, where $z_{B,j} = \max_i a(W_j^\top h_{B,i})$, the output depends only on the maximally activated instance $i^*_{B,j}$. The derivatives are:

$$\frac{\partial z_B}{\partial w_j} = u_{B,i^*_{B,j},j}, \quad \frac{\partial z_B}{\partial W_j} = w_j \, h_{B,i^*_{B,j}} \, \mathbf{1}\{W_j^\top h_{B,i^*_{B,j}} > 0\}.$$

Thus, $\gamma_{B,j}^{\max} = u^2_{B,i^*_{B,j},j} + w_j^2 \mathbf{1}\left\{ W_j h_{B,i^*_B(j)} > 0 \right\} \left\| h_{B,i^*_B(j)} \right\|^2$.

For **mean Pooling**, where $z_{B,j} = \frac{1}{T_{\mathbf{B}}} \sum_{i=1}^{T_{\mathbf{B}}} a(W_j^\top h_{B,i})$, the derivatives are averaged over the bag:

$$\frac{\partial z_B}{\partial w_j} = \frac{1}{T_{\mathbf{B}}} \sum_{i=1}^{T_{\mathbf{B}}} u_{B,i,j}, \quad \frac{\partial z_B}{\partial W_j} = \frac{w_j}{T_{\mathbf{B}}} \sum_{i=1}^{T_{\mathbf{B}}} h_{B,i} \, \mathbf{1}\{W_j^\top h_{B,i} > 0\}.$$

This yields $\gamma_{B,j}^{\text{mean}} = \left(\frac{1}{T_{\mathbf{B}}} \sum_{i=1}^{T_{\mathbf{B}}} u_{B,i,j} \right)^2 + \frac{w_j^2}{T_{\mathbf{B}}^2} \left\| \sum_{i=1}^{T_{\mathbf{B}}} h_{B,i} \, \mathbf{1}\{W_j^\top h_{B,i} > 0\} \right\|_2^2$.

Combining these with the margin update equation completes the proof.

B.4 Proof of Theorem 2

For a fixed starting parameter θ, define $\theta_{\mathbf{B},\eta} := \theta - \eta\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta)$ and $\mathcal{M}_\eta(\theta) := \mathbb{E}_{\mathbf{B}}[m_{\mathbf{B}}(\theta_{\mathbf{B},\eta})]$, as in Theorem 2. The expectation is over the finite empirical bag distribution, with $\mathcal{M}_0(\theta) = \mathcal{M}(\theta)$. Each term evaluates the same bag that induces its update, starting from the same θ.

$z_{\mathbf{B}}$ is piecewise affine in its input for fixed weights. Jointly in $\theta = (W, w)$ it is piecewise polynomial and smooth inside regions with fixed ReLU and max-pooling selections. We take a sufficiently small update within such a region. The derivative with respect to η at $\eta = 0$ is:

$$\frac{\mathrm{d}}{\mathrm{d}\eta} m_{\mathbf{B}}(\theta_{\mathbf{B},\eta})\bigg|_{\eta=0} = -\nabla_\theta m_{\mathbf{B}}(\theta)^\top \nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta).$$

Substituting the gradient $\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta) = -\frac{y_{\mathbf{B}}}{1+\exp(m_{\mathbf{B}}(\theta))}\nabla_\theta z_{\mathbf{B}}(\theta)$:

$$\begin{aligned}-\nabla_\theta m_{\mathbf{B}}(\theta)^\top \nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta) &= -(y_{\mathbf{B}}\nabla_\theta z_{\mathbf{B}}(\theta))^\top \left(-\frac{y_{\mathbf{B}}}{1+\exp(m_{\mathbf{B}}(\theta))}\nabla_\theta z_{\mathbf{B}}(\theta)\right) \\ &= \frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1+\exp(m_{\mathbf{B}}(\theta))}.\end{aligned}$$

For the finite empirical bag distribution, differentiation commutes with averaging:

$$\frac{\mathrm{d}}{\mathrm{d}\eta}\mathcal{M}_\eta(\theta)\bigg|_{\eta=0} = \mathbb{E}_{\mathbf{B}}\left[\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1+\exp(m_{\mathbf{B}}(\theta))}\right] \geq 0.$$

The Taylor expansion around $\eta = 0$ gives:

$$\mathcal{M}_\eta(\theta) = \mathcal{M}_0(\theta) + \eta\mathbb{E}_{\mathbf{B}}\left[\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1+\exp(m_{\mathbf{B}}(\theta))}\right] + O(\eta^2).$$

For sufficiently small $\eta > 0$, $\mathcal{M}_\eta(\theta) \geq \mathcal{M}_0(\theta)$, with strict increase when the expected first-order term is positive. Each bag is evaluated after its own update; this is not a statement about the population margin after one shared update.

B.5 Proof of Theorem 3

Let $J_{\mathbf{B}} := \{j : s_{\mathbf{B},j} \geq 1\}$ be the set of active channels. For $j \notin J_{\mathbf{B}}$, the gradient contribution $\gamma_{\mathbf{B},j}$ vanishes by Assumption 1. Thus, the squared norms decompose as $\sum_{j\in J_{\mathbf{B}}} \gamma_{\mathbf{B},j}$.

1. Max Pooling Lower Bound. For $j \in J_{\mathbf{B}}$, let $i^* \in S_{\mathbf{B},j}$ be the maximizing instance. By Assumption 1, we have:

$$\gamma^{\max}_{\mathbf{B},j} = u^2_{\mathbf{B},i^*,j} + w_j^2\|\mathbf{g}_{\mathbf{B},i^*,j}\|_2^2 \geq c_1(1+w_j^2),$$

where $c_1 := \min\{\underline{u}^2, \underline{g}^2\}$. Summing over $J_{\mathbf{B}}$ yields $\|\nabla_\theta z^{\max}_{\mathbf{B}}\|^2 \geq c_1 \sum_{j\in J_{\mathbf{B}}}(1+w_j^2)$.

2. Mean Pooling Upper Bound. For mean pooling, since only $i \in S_{\mathbf{B},j}$ are non-zero ($s_{\mathbf{B},j} \leq s_{\mathbf{B}}$), the triangle inequality and Assumption 1 give:

$$\gamma^{\mathrm{mean}}_{\mathbf{B},j} = \left(\frac{\sum_{i\in S_{\mathbf{B},j}} u_{i,j}}{T_{\mathbf{B}}}\right)^2 + \frac{w_j^2}{T_{\mathbf{B}}^2}\left\|\sum_{i\in S_{\mathbf{B},j}} \mathbf{g}_{i,j}\right\|_2^2 \leq \frac{s_{\mathbf{B}}^2}{T_{\mathbf{B}}^2}c_2(1+w_j^2),$$

where $c_2 := \max\{\overline{u}^2, \overline{g}^2\}$. Thus, $\|\nabla_\theta z^{\mathrm{mean}}_{\mathbf{B}}\|^2 \leq c_2(\frac{s_{\mathbf{B}}}{T_{\mathbf{B}}})^2 \sum_{j\in J_{\mathbf{B}}}(1+w_j^2)$.

3. Conclusion. Combining these bounds, we obtain:

$$\frac{\|\nabla_\theta z^{\max}_{\mathbf{B}}\|^2}{\|\nabla_\theta z^{\mathrm{mean}}_{\mathbf{B}}\|^2} \geq \frac{c_1}{c_2}\left(\frac{T_{\mathbf{B}}}{s_{\mathbf{B}}}\right)^2 = \Omega\left(\left(\frac{T_{\mathbf{B}}}{s_{\mathbf{B}}}\right)^2\right).$$

B.6 Proof of Proposition 1

We provide the derivations for the Rademacher complexity bounds of both the feature-extraction-based model ($\mathcal{F}_{\text{feat}}$) and the baseline model ($\mathcal{F}_{\text{base}}$).

1. Bound for the Model with Feature Extraction Layer ($\mathcal{F}_{\text{feat}}$) We first bound the empirical Rademacher complexity of the hypothesis class:

$$\mathcal{F}_{\text{feat}} = \left\{ f_\theta(\mathbf{B}) = w^\top \rho\big(\{a(W_j^\top h_t)\}_{j,t}\big) \;\middle|\; \|W_j\|_2 \le B_1, \|w\|_2 \le B_2 \right\}.$$

Applying the Cauchy-Schwarz inequality to separate the classification weights w, we have:

$$\hat{\mathcal{R}}_S(\mathcal{F}_{\text{feat}}) \le \frac{B_2}{n} \mathbb{E}_\sigma \left[\sup_{\|W_j\|_2 \le B_1} \left\| \sum_{i=1}^n \sigma_i \rho(\{a(W^\top h_{i,t})\}_t) \right\|_2 \right].$$

Since the norm constraint $\|W_j\|_2 \le B_1$ applies independently to each column $j \in \{1, \dots, D\}$, the supremum of the L_2 norm factors into independent suprema across the D dimensions:

$$\sup_W \left\| \sum_{i=1}^n \sigma_i \rho(\{a(W^\top h_{i,t})\}_t) \right\|_2 \le \sqrt{D} \sup_{\|w_1\|_2 \le B_1} \left| \sum_{i=1}^n \sigma_i \max_{1 \le t \le T_i} a(w_1^\top h_{i,t}) \right|.$$

By introducing a factor of 2 to handle the absolute value, and sequentially applying Maurer's vector contraction inequality (Maurer, 2016) for the max operator which introduces independent Rademacher variables $\sigma_{i,t} \in \{-1, +1\}$ for each instance t and the Ledoux-Talagrand contraction inequality for the ReLU function $a(\cdot)$, we obtain:

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathcal{F}_{\text{feat}}) &\le \frac{2\sqrt{2} B_2 \sqrt{D}}{n} \mathbb{E}_\sigma \left[\sup_{\|w_1\|_2 \le B_1} \sum_{i=1}^n \sum_{t=1}^{T_i} \sigma_{i,t} w_1^\top h_{i,t} \right] \\
&\le \frac{2\sqrt{2} B_1 B_2 \sqrt{D}}{n} \mathbb{E}_\sigma \left[\left\| \sum_{i=1}^n \sum_{t=1}^{T_i} \sigma_{i,t} h_{i,t} \right\|_2 \right] \\
&\le \frac{2\sqrt{2} R B_1 B_2 \sqrt{DT}}{\sqrt{n}},
\end{aligned}$$

where $T = \max_i T_i$. This gives $\mathcal{O}(\sqrt{DT/n})$ scaling with no explicit dependence on d under the stated norm bounds.

2. Bound for the Baseline Model ($\mathcal{F}_{\text{base}}$) For the baseline class $\mathcal{F}_{\text{base}}$, max pooling $\rho(\mathbf{B}_i)$ is performed directly on the d-dimensional input space. Let $z_i = \rho(\mathbf{B}_i) \in \mathbb{R}^d$. Under the assumption $\|h_{i,t}\|_2 \le R$, which implies $|h_{i,t,\ell}| \le R$ for each coordinate, the norm of the pooled vector is bounded by:

$$\|z_i\|_2 \le \sqrt{\sum_{k=1}^d (\max_t |h_{i,t,k}|)^2} \le \sqrt{dR^2} = R\sqrt{d}.$$

Treating this as a standard two-layer neural network with weight constraints B_1, B_2 acting on $\{z_i\}$, we apply standard Rademacher complexity results:

$$\begin{aligned}
\hat{\mathcal{R}}_S(\mathcal{F}_{\text{base}}) &\le \frac{2B_1 B_2 \sqrt{D}}{n} \mathbb{E}_\sigma \left\| \sum_{i=1}^n \sigma_i z_i \right\|_2 \\
&\le \frac{2B_1 B_2 \sqrt{D}}{\sqrt{n}} \max_i \|z_i\|_2 \le \frac{2RB_1 B_2 \sqrt{Dd}}{\sqrt{n}}.
\end{aligned}$$

This establishes the displayed coordinate-wise bound. A sharper estimate uses

$$\|z_i\|_2^2 \le \sum_{k=1}^d \max_t |h_{i,t,k}|^2 \le \sum_{t=1}^{T_i} \|h_{i,t}\|_2^2 \le T_i R^2.$$

Combining the estimates gives $\hat{\mathcal{R}}_S(\mathcal{F}_{\text{base}}) \le 2RB_1 B_2 \sqrt{D \min(d, T)/n}$. Thus, these norm-based upper bounds alone do not establish a smaller complexity for feature-first pooling. The feature-placement ablation in Appendix F.1 supplies the empirical comparison.

B.7 Margin Expansion under Per-Bag Updates and β-Smoothness

Theorem 2 describes the average change in each bag's margin under its own update, $\mathcal{M}_\eta(\theta) - \mathcal{M}_0(\theta)$. Its first-order Taylor expansion is local in η: the $O(\eta^2)$ remainder can dominate the positive first-order term. We use a β-smoothness assumption to bound this remainder and specify a sufficient step-size condition for this average to increase.

Assumption 2 (β-Smoothness of the Margin)**.** *For any sampled bag $\mathbf{B}$, assume the margin function $m_{\mathbf{B}}(\theta)$ is β-smooth on a neighborhood containing its update segment. ReLU and max pooling do not guarantee global smoothness across selection boundaries. That is, its gradient is Lipschitz continuous with a constant $\beta > 0$, implying that for parameters θ and update steps $\Delta\theta$ within that neighborhood:*

$$m_{\mathbf{B}}(\theta + \Delta\theta) \geq m_{\mathbf{B}}(\theta) + \nabla_\theta m_{\mathbf{B}}(\theta)^\top \Delta\theta - \frac{\beta}{2}\|\Delta\theta\|^2.$$

For the logistic-loss update induced by a sampled bag $\mathbf{B}$, let $\Delta\theta = -\eta\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta)$ and $\theta_{\mathbf{B},\eta} := \theta + \Delta\theta$. Substituting this bag-specific update into the β-smoothness inequality yields:

$$m_{\mathbf{B}}(\theta_{\mathbf{B},\eta}) - m_{\mathbf{B}}(\theta) \geq -\eta\nabla_\theta m_{\mathbf{B}}(\theta)^\top \nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta) - \frac{\beta\eta^2}{2}\|\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta)\|^2.$$

Recall from Appendix B.4 that $\nabla_\theta m_{\mathbf{B}}(\theta) = y_{\mathbf{B}}\nabla_\theta z_{\mathbf{B}}(\theta)$ and $\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta) = -\frac{y_{\mathbf{B}}}{1+\exp(m_{\mathbf{B}}(\theta))}\nabla_\theta z_{\mathbf{B}}(\theta)$. Since $y_{\mathbf{B}} \in \{-1, 1\}$, the squared norm of the loss gradient simplifies to:

$$\|\nabla_\theta \mathcal{L}_{\mathbf{B}}(\theta)\|^2 = \frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{(1 + \exp(m_{\mathbf{B}}(\theta)))^2}.$$

Substituting the gradient terms back into the inequality gives:

$$\begin{aligned} m_{\mathbf{B}}(\theta_{\mathbf{B},\eta}) - m_{\mathbf{B}}(\theta) &\geq \eta\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1 + \exp(m_{\mathbf{B}}(\theta))} - \frac{\beta\eta^2}{2}\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{(1 + \exp(m_{\mathbf{B}}(\theta)))^2} \\ &= \eta\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1 + \exp(m_{\mathbf{B}}(\theta))}\left(1 - \frac{\beta\eta}{2(1 + \exp(m_{\mathbf{B}}(\theta)))}\right). \end{aligned}$$

Taking the expectation over the finite empirical bag distribution bounds the average change in each bag's own margin:

$$\mathcal{M}_\eta(\theta) - \mathcal{M}_0(\theta) \geq \mathbb{E}_{\mathbf{B}}\left[\eta\frac{\|\nabla_\theta z_{\mathbf{B}}(\theta)\|^2}{1 + \exp(m_{\mathbf{B}}(\theta))}\left(1 - \frac{\beta\eta}{2(1 + \exp(m_{\mathbf{B}}(\theta)))}\right)\right].$$

If at least one bag has a nonzero gradient, a sufficient condition for $\mathcal{M}_\eta(\theta) > \mathcal{M}_0(\theta)$ is that the factor in parentheses is positive for every bag and every update stays in its smooth neighborhood. The step-size condition is:

$$0 < \eta < \frac{2(1 + \exp(m_{\mathbf{B}}(\theta)))}{\beta} \quad \text{for every } \mathbf{B}.$$

Since $\exp(m_{\mathbf{B}}(\theta)) > 0$, we have $1 + \exp(m_{\mathbf{B}}(\theta)) > 1$. Therefore, a sufficient condition for the learning rate to guarantee margin expansion, independent of the current margin value, is:

$$0 < \eta \leq \frac{2}{\beta}.$$

Provided each update remains in its smooth neighborhood, this condition guarantees an increase in each bag's own margin whenever its gradient is nonzero, and hence in their average when at least one bag has a nonzero gradient.

C GENERALIZATION TO NON-INVARIANT ACTIVE SET IN HAMI

In the analysis of ReLU networks, the fact that input perturbations cause changes in activation patterns poses a significant challenge. In this section, we show that the entire network is a Continuous Piecewise-Affine (CPWA) map, mathematically guaranteeing the validity of sensitivity analysis through path integration.

C.1 PROPERTIES AND COMPOSITIONS OF PIECEWISE-AFFINE MAPS

Here, we summarize the properties regarding the composition of piecewise-affine maps defined on a finite number of polyhedral regions.

Definition: Piecewise-Affine Map A function $F : \mathbb{R}^d \to \mathbb{R}^m$ is defined as piecewise-affine if there exists a partition of the input space $R_1, \ldots, R_M$, such that on each region R_i, it can be described using $A_i \in \mathbb{R}^{m\times d}$ and $b_i \in \mathbb{R}^m$ as follows:

$$F(x) = A_i x + b_i \quad (\forall x \in R_i).$$

Primary Properties of Composition The following properties hold for the composition of affine maps and piecewise-affine maps. Furthermore, since the composition of continuous functions is continuous, if each component is continuous, the composite function is also a continuous piecewise-affine (CPWA) map.

1. **(Affine)** $\circ$ **(Affine)** is an affine map.
2. **(Affine)** $\circ$ **(PWA)** is a piecewise-affine map.
3. **(PWA)** $\circ$ **(Affine)** is a piecewise-affine map.
4. **(PWA)** $\circ$ **(PWA)** is a piecewise-affine map.

Mathematical Proofs

- **Proof of 1:** Let $H(x) := Ax+b$ and $G(y) := Cy+d$. Then $(G\circ H)(x) = C(Ax+b)+d = (CA)x + (Cb + d)$, which is an affine map.
- **Proof of 2:** Let F be an affine map $A_i x + b_i$ on region R_i. Then $(G \circ F)(x) = C(A_i x + b_i) + d = (CA_i)x + (Cb_i + d)$, making it a PWA map on the same region R_i.
- **Proof of 3:** Let G be an affine map $C_j y+d_j$ on region S_j. The inverse image $T_j = H^{-1}(S_j)$ is also a polyhedral region, and on each T_j, the map becomes $(C_j A)x + (C_j b + d_j)$, thus it is a PWA map.
- **Proof of 4:** Let the regions for F and G be R_i and S_j, respectively. Defining the region where the composite function is affine as $T_{ij} = R_i \cap \{x : A_i x + b_i \in S_j\}$, this forms a polyhedral region as the intersection of polyhedral regions. On each T_{ij}, it can be written as $C_j(A_i x + b_i) + d_j$, meaning it is a PWA map.

C.2 ANALYSIS OF CONTINUOUS PIECEWISE-AFFINE PROPERTIES IN RELU NETWORKS (ARORA ET AL., 2018)

A function $F : \mathbb{R}^d \to \mathbb{R}^m$ is defined as **continuous piecewise-affine (CPWA)** if the input space is partitioned into a finite number of polyhedral regions $R_1, \ldots, R_L$, and on each region R_ℓ, it can be described as an affine map $F(x) = A_\ell x + b_\ell$ using $A_\ell \in \mathbb{R}^{m\times d}$ and $b_\ell \in \mathbb{R}^m$.

In a ReLU network, each region R_ℓ corresponds to a region where the **active pattern** of all ReLU units is fixed. That is, within a certain region, the On/Off state of each unit is invariant, and the ReLU can be viewed as a linear transformation Du using a diagonal mask matrix $D = \mathrm{diag}(\mathbb{1}_{\{u>0\}})$ (Montúfar et al., 2014). For example, in a two-layer ReLU network $F(x) = A_2\,\mathrm{ReLU}(A_1x+b_1)+b_2$, since $D_1 = \mathrm{diag}(\mathbb{1}_{\{A_1x+b_1>0\}})$ is fixed within a specific activation region,

$$F(x) = A_2 D_1(A_1 x + b_1) + b_2 = A_2 D_1 A_1 x + (A_2 D_1 b_1 + b_2)$$

which reduces to an affine map $A'x + c$.

Generalizing this to an L-layer deep ReLU network, let the output of each layer be $h_0 = x$, $h_\ell = \mathrm{ReLU}(A_\ell h_{\ell-1} + b_\ell)$ for $\ell = 1, \ldots, L-1$, and the final output be $F(x) = A_L h_{L-1} + b_L$. Within a region where a certain activation pattern $D = (D_1, \ldots, D_{L-1})$ is fixed, it can be written as follows:

$$F(x) = (A_L D_{L-1} A_{L-1} \cdots D_1 A_1)\, x + c(D)$$

Here, $c(D)$ is a constant term that depends only on the activation pattern and is independent of the input x within the region:

$$c(D) = b_L + A_L D_{L-1} b_{L-1} + A_L D_{L-1} A_{L-1} D_{L-2} b_{L-2} + \cdots + A_L D_{L-1} \cdots A_2 D_1 b_1.$$

Thus, a ReLU network partitions the input space into numerous polyhedral regions and possesses the structure of a CPWA function that behaves locally linearly within each region.

C.3 Sensitivity Analysis via Path Integration of Continuous Piecewise-Affine Functions (Sundararajan et al., 2017)

Let the function $F : \mathbb{R}^d \to \mathbb{R}$ be a continuous piecewise-affine (CPWA) function described as $F(x) = a_m^\top x + b_m$ on a finite number of polyhedral regions $R_1, \ldots, R_M$. As discussed previously, deep ReLU networks also reduce to this form. In this subsection, we derive the change in output for a finite perturbation from an input $x_0 \in \mathbb{R}^d$ to $x_1 = x_0 + \Delta x$ as the integral of local sensitivities along a path.

First, we define a linear path $\gamma(t) := x_0 + t\Delta x$ (where $t \in [0, 1]$) connecting the two points, and consider a single-variable function $g(t) := F(\gamma(t))$ along this path. Even if F is a multivariable CPWA function, $g(t)$ becomes an affine map with respect to t on each interval, thus forming a continuous piecewise-linear (CPWL) function on $[0, 1]$. Specifically, when the path is within a region R_m,

$$g(t) = a_m^\top(x_0 + t\Delta x) + b_m = (a_m^\top \Delta x)t + (a_m^\top x_0 + b_m)$$

and its slope changes each time the path enters a different region. Therefore, letting the points where the path crosses region boundaries be $0 = \tau_0 < \tau_1 < \cdots < \tau_M = 1$, $g(t)$ can be written as $g(t) = \alpha_m t + \beta_m$ in each interval (τ_{m-1}, τ_m). Applying the fundamental theorem of calculus in each interval and summing over all intervals yields:

$$g(1) - g(0) = \sum_{m=1}^{M} [g(\tau_m) - g(\tau_{m-1})] = \sum_{m=1}^{M} \int_{\tau_{m-1}}^{\tau_m} g'(t)\mathrm{d}t = \int_0^1 g'(t)\mathrm{d}t.$$

Since the breakpoints τ_m are finite in number (measure zero), they do not affect the integral value. Here, by the chain rule, at points where F is differentiable along $\gamma(t)$, $g'(t) = \nabla F(\gamma(t))^\top \gamma'(t) = \nabla F(x_0 + t\Delta x)^\top \Delta x$, finally leading to the following equation:

$$F(x_1) - F(x_0) = \int_0^1 \nabla F(x_0 + t\Delta x)^\top \Delta x \mathrm{d}t.$$

Viewing this result in the form of a finite sum gives $F(x_1) - F(x_0) = \sum_{m=1}^{M} (\tau_m - \tau_{m-1}) a_m^\top \Delta x$, which explicitly shows the process where the local sensitivity $a_m^\top \Delta x$ obtained in each region accumulates according to the residence time.

C.4 Application to HaMI: Semantic Scaling and Integral Sensitivity Analysis

In this section, we apply the theoretical foundations established so far to the semantic scaling of HaMI under inference-time. First, we show that the bag-level logit score $Z_{\mathbf{B}}(p) = \mathrm{TopK\text{-}Mean}_i\, f((1 + p)h_{\mathbf{B},i})$ when scaling the hidden state as $h(p) = (1 + p)h$ preserves the CPWA property.

To this end, we prove that the aggregation operation $\mathrm{TopKSum}$ is a CPWA map. For two continuous functions f and g, their maximum can be written as $\max\{f(x), g(x)\} = \frac{1}{2}(f(x) + g(x) + |f(x) - g(x)|)$. Because the sum, difference, absolute value, and composition of continuous functions are continuous, the maximum of a finite number of continuous functions is also continuous. The sum of

the top k elements for a vector $u \in \mathbb{R}^T$ can be defined using the linear form $a_S^\top u$ for the index set $S \subseteq \{1, \ldots, T\}$ with $|S| = k$ as:

$$\mathrm{TopKSum}_k(u) = \max_{S:|S|=k} a_S^\top u$$

This is an operation that takes the maximum of a finite number of linear functions (Max-affine); it is affine on each region $R_S = \{u : a_S^\top u \geq a_{S'}^\top u, \forall S'\}$ and continuous overall. Therefore, $\mathrm{TopKSum}$ and its scalar multiple, $\mathrm{TopK\text{-}Mean}$, are CPWA maps.

From the above discussion, even if the model f contains ReLU, Batch Normalization, and various pooling operations, their composition $Z_{\mathbf{B}}(p)$ becomes a single-variable continuous piecewise-linear (CPWL) function with respect to p. Based on this property, applying the sensitivity analysis from the previous section allows the change in output to be described as follows:

$$Z_{\mathbf{B}}(p_{\mathbf{B}}) - Z_{\mathbf{B}}(0) = \int_0^{p_{\mathbf{B}}} Z'_{\mathbf{B}}(p)\mathrm{d}p,$$
$$Z_{\mathbf{B}}(p_{\mathbf{B}}) = Z_{\mathbf{B}}(0) - p_{\mathbf{B}}\bar{C}^{\mathrm{int}}_{\mathbf{B}}(p_{\mathbf{B}}).$$

Here, $\bar{C}^{\mathrm{int}}_{\mathbf{B}}(p_{\mathbf{B}}) := -\frac{1}{p_{\mathbf{B}}}\int_0^{p_{\mathbf{B}}} Z'_{\mathbf{B}}(p)\mathrm{d}p$ is the average cost term calculated across changes in the active set. Furthermore, by setting $p_{\mathbf{B}} = \lambda P^{\mathbf{B}}_{\mathrm{sem}}$, the general formula for HaMI is obtained:

$$Z_{\mathbf{B}}(\lambda P^{\mathbf{B}}_{\mathrm{sem}}) = Z_{\mathbf{B}}(0) - \lambda P^{\mathbf{B}}_{\mathrm{sem}}\bar{C}^{\mathrm{int}}_{\mathbf{B}}(\lambda P^{\mathbf{B}}_{\mathrm{sem}}).$$

This formulation maintains mathematical consistency even under dynamic situations where the activation pattern changes.

Numerical Approximation: In our empirical evaluations, we set λ to 1 and approximate the integral by discretizing the interval $p \in [0, p_{\mathbf{B}}]$ into $N = 1000$ uniform steps. For each step k, we compute the local gradient $Z'_{\mathbf{B}}(p_k)$. This high-resolution discretization provides an accurate numerical approximation of the integrated sensitivity, even when the active set changes frequently.

D Data and Label Validation

Shared generation and filtering procedures are described in Appendix A.1.

D.1 Short-answer QA

Counts and class balance. Table 10 reports class balance for the four QA benchmarks on the training and test splits, after the two-pass label filter of Appendix D.3. Labels follow the convention of Section 6.1: the positive class is a hallucinated generation ($1 - \mathrm{accuracy} = 1$). Five responses are generated per question, but detector training uses only one and test uses all five. The splits have different question counts, and N also reflects filtering (Appendix A.1).

On the short-answer benchmarks the balance shifts systematically with generator strength: the negative-to-positive ratio on TriviaQA rises from 3.26 (LLaMA-3.1-8B) to 4.69 (Mistral-Nemo-12B) to 12.21 (LLaMA-3.3-70B). LongFact class counts are reported separately for each generation limit in Table 12.

The class ratio enters Theorem 1. Semantic scaling enlarges the expected margin only when $|\mathcal{S}_{neg}|/|\mathcal{S}_{pos}| > 1/\gamma$, and the values of $1/\gamma$ we measure on the short-answer benchmarks lie between 0.38 and 0.51 (Appendix G.1). Every short-answer ratio in Table 10 clears that band, the smallest being 0.52 for LLaMA-3.1-8B on SQuAD, so the condition holds wherever we have checked it.

Table 10: QA class counts after filtering (N). "Halluc." is positive and "Correct" negative; Neg/Pos is their negative-to-positive ratio. Of five responses generated per question, training uses one and test uses all five.

		Train				Test			
Model	**Dataset**	N	Halluc.	Correct	Neg/Pos	N	Halluc.	Correct	Neg/Pos
LLaMA-3.1-8B	TriviaQA	3,870	996	2,874	**2.89**	5,354	1,257	4,097	**3.26**
	SQuAD	3,290	2,163	1,127	**0.52**	4,480	2,957	1,523	**0.52**
	NQ	3,606	2,222	1,384	**0.62**	4,968	3,151	1,817	**0.58**
	BioASQ	2,141	1,081	1,060	**0.98**	2,766	1,334	1,432	**1.07**
Mistral-Nemo-12B	TriviaQA	3,866	705	3,161	**4.48**	5,329	937	4,392	**4.69**
	SQuAD	3,154	1,810	1,344	**0.74**	4,274	2,449	1,825	**0.75**
	NQ	3,469	1,626	1,843	**1.13**	4,751	2,356	2,395	**1.02**
	BioASQ	2,054	765	1,289	**1.68**	2,687	1,035	1,652	**1.60**
LLaMA-3.3-70B	TriviaQA	3,875	343	3,532	**10.30**	5,362	406	4,956	**12.21**
	SQuAD	3,046	1,353	1,693	**1.25**	4,114	1,826	2,288	**1.25**
	NQ	3,349	1,040	2,309	**2.22**	4,531	1,462	3,069	**2.10**
	BioASQ	2,006	603	1,403	**2.33**	2,629	784	1,845	**2.35**

Response length. Figure 3 plots response lengths for the four QA benchmarks and two LongFact generation limits. Table 11 summarizes QA lengths; Table 13 summarizes LongFact. Counts are read from the stored per-token log-likelihoods, so they count exactly the positions a pooling detector aggregates over — that is, the bag size $T_{\mathbf{B}}$ of Section 5.1.

The four short-answer benchmarks were prompted for a single sentence, and their distributions are correspondingly short and right-skewed, with medians between 4 and 33 tokens and almost all mass below 60. Length nonetheless varies widely between generators under an identical prompt, though not monotonically in model size: LLaMA-3.3-70B answers TriviaQA in 24.5 tokens on average against 8.7 for LLaMA-3.1-8B and 5.5 for Mistral-Nemo-12B, so the bags reaching the detector differ in size by roughly a factor of three across generators on the same questions.

LongFact uses the length-specific prompt and sentence-boundary trimming in Appendix D.2. Stored lengths are measured after trimming and do not identify every generation that reached the token limit.

Table 11: Retained QA response lengths in tokens, counted from stored per-token log-likelihoods.

		Train			Test		
Model	**Dataset**	Mean ± SD	Median	p_{95}	Mean ± SD	Median	p_{95}
LLaMA-3.1-8B	TriviaQA	8.7 ± 11.4	5	28	8.7 ± 11.7	5	28
	SQuAD	16.8 ± 15.9	13	48	17.7 ± 17.2	13	53
	NQ	18.5 ± 18.3	13	58	19.3 ± 18.7	14	60
	BioASQ	18.8 ± 20.1	12	62	18.5 ± 20.4	11	63
Mistral-Nemo-12B	TriviaQA	6.0 ± 5.2	4	17	5.5 ± 4.5	4	15
	SQuAD	12.2 ± 8.1	10	27	12.3 ± 8.3	10	28
	NQ	13.2 ± 9.2	12	29	13.8 ± 9.8	12	31
	BioASQ	16.0 ± 9.5	15	32	15.8 ± 9.3	15	33
LLaMA-3.3-70B	TriviaQA	24.7 ± 13.6	23	48	24.5 ± 13.3	23	46
	SQuAD	26.8 ± 13.4	25	49	27.5 ± 13.7	25	52
	NQ	26.8 ± 12.8	24	48	28.1 ± 13.7	25	52
	BioASQ	34.5 ± 15.6	32	60	34.6 ± 15.4	33	61

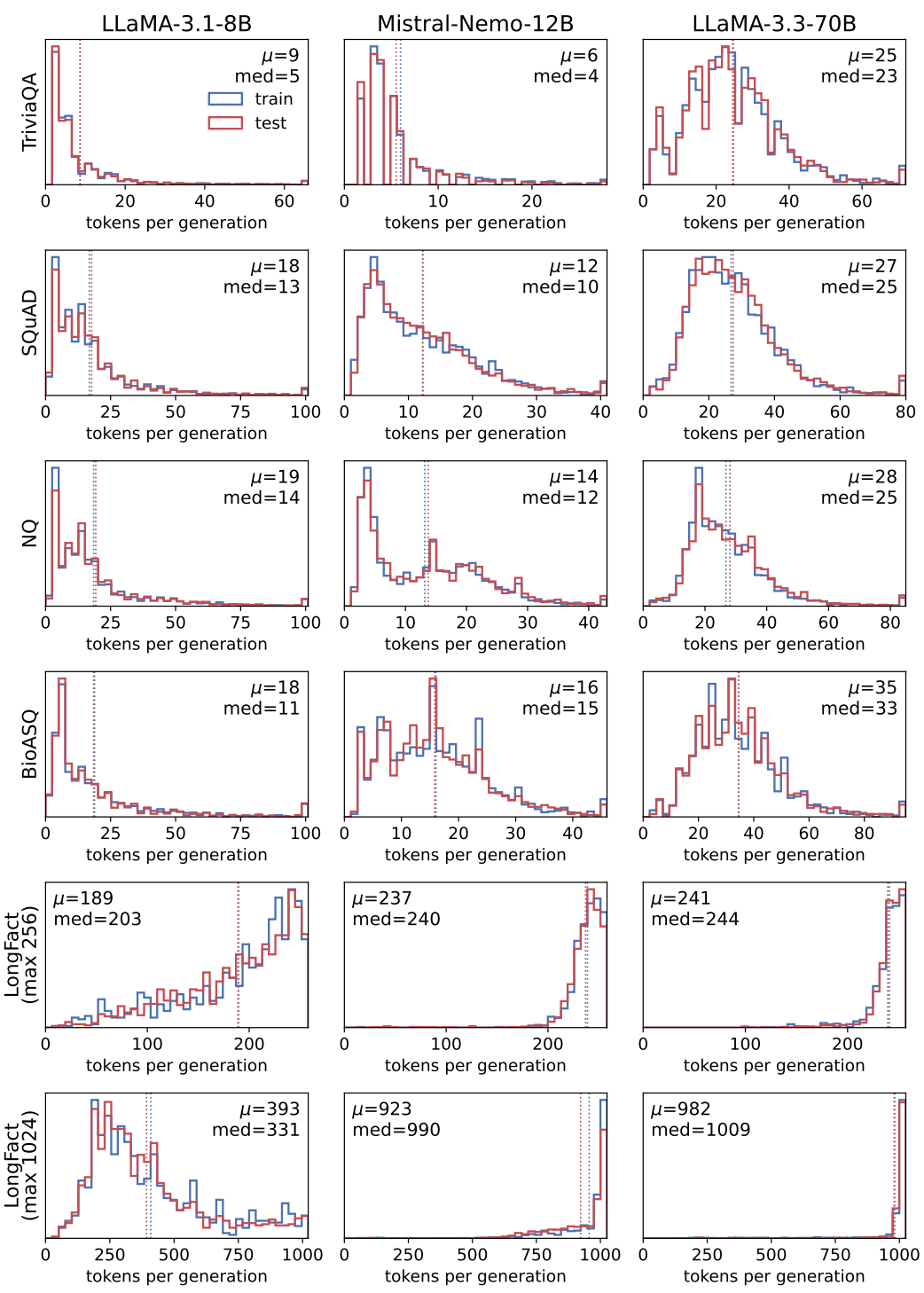


Figure 3: Response-length distributions by dataset (rows) and model (columns). Histograms have unit area; dotted lines mark means. Annotations give test-set means and medians. Panel-specific axes extend to the 99th percentile. LongFact uses 256- and 1024-token limits.

D.2 LONGFACT

LongFact uses its own prompts and has no reference answers. Its responses are substantially longer than those of the QA benchmarks. We vary the generation limit while keeping each method's selected layer fixed across settings.

Generation setting. Raising `max_new_tokens` alone does not lengthen the answers, so each limit requests its length in the prompt (Prompt 2 in Appendix A.1), with (W, K) set to $(180, 3)$ at 256 tokens and $(750, 10)$ at 1024. LLaMA-3.1-8B is a base model and does not follow that instruction, so its prompt ends with the start of the enumeration. Responses are generated with vLLM (Kwon et al., 2023), with an AWQ (Lin et al., 2024) 4-bit checkpoint for LLaMA-3.3-70B. A teacher-forcing pass over the saved text extracts hidden states. A numerical comparison with generation-time states gave cosine similarity above 0.999; this is an agreement check rather than an assertion of exact equivalence between the two implementations.

Asking for length also changes factual content: the hallucination rate of the two instruction-tuned models falls from 83.6% and 72.8% under the unmodified prompt to 12.3% and 14.8% at 256 tokens, while the base model, which does not follow the instruction, barely moves (92.9% to 90.4%). These cells are therefore not comparable with a run that leaves the prompt unchanged.

Counts and class balance. Table 12 counts generations, not questions. Training keeps one response per question and validation and test keep all five, so the design sizes are $1{,}000$, $1{,}000$ and $1{,}500$; the shortfall is the two-pass filter of Appendix D.3. At the 1024-token limit, Mistral-Nemo-12B and LLaMA-3.3-70B have 69 and 35 hallucinated training responses, respectively, so the positive class is sparsely represented in those training splits.

Instruction tuning may contribute to these differences in class balance. The two Instruct models respond more strongly to the length-and-structure prompt than the base model, potentially altering which factual claims are generated. This interpretation remains provisional because model family, size and pretraining also differ, and LongFact labels depend on GPT-5 mini's internal knowledge.

Table 12: Retained LongFact responses (hallucinated counts in parentheses). Training uses one response per question; validation and test use all five.

Model	Setting	Train	Validation	Test
LLaMA-3.1-8B	LongFact (max 256)	956 (864)	950 (802)	1,415 (1,171)
	LongFact (max 1024)	956 (872)	954 (815)	1,424 (1,261)
Mistral-Nemo-12B	LongFact (max 256)	936 (115)	933 (160)	1,428 (285)
	LongFact (max 1024)	986 (69)	976 (110)	1,464 (149)
LLaMA-3.3-70B	LongFact (max 256)	934 (138)	948 (105)	1,408 (186)
	LongFact (max 1024)	981 (35)	988 (46)	1,471 (75)

Response length. Table 13 reports the length of each retained response, counted as in Appendix D.1; Figure 3 plots the distributions alongside the QA benchmarks.

Table 13: Retained LongFact response lengths in tokens. Evaluation combines validation and test; "Limit" denotes the maximum number of new tokens.

		Train			Validation + test		
Model	**Limit**	Mean ± SD	Median	p_{95}	Mean ± SD	Median	p_{95}
LLaMA-3.1-8B	256	189.7 ± 58.5	209	253	187.9 ± 55.5	201	252
	1024	409.3 ± 233.5	340	935	391.1 ± 220.2	329	888
Mistral-Nemo-12B	256	238.8 ± 16.0	241	255	236.8 ± 22.0	241	255
	1024	956.2 ± 105.0	1003	1023	922.4 ± 134.3	989	1022
LLaMA-3.3-70B	256	239.4 ± 18.9	244	255	240.8 ± 17.1	245	255
	1024	979.4 ± 123.4	1009	1023	982.9 ± 113.7	1009	1023

Layer selection. A full layer search with a generation budget above $1{,}000$ tokens is computationally demanding. We therefore select each method's layer by validation AUC using responses generated with at most 256 new tokens, and keep that layer fixed for both the 256- and 1024-token settings. This selection is performed separately from the QA layer search in Appendix A.4 because the label distribution differs.

Table 14: Layers selected by validation AUC at the 256-token limit and reused at 1024 tokens.

Method	LLaMA-3.1-8B	Mistral-Nemo-12B	LLaMA-3.3-70B
SEP	28	6	0
LLMs Know (last token)	30	12	48
SAPLMA	28	12	48
HaMI (Original)	14	12	28
HaMI (SP)	14	12	28
Mean Pooling	18	18	36
Max Pooling	14	8	76

D.3 Label Validation

Agreement with human annotation. Our detectors are trained against labels produced by an LLM judge (Section 6.1) rather than by human annotators. A probe fitted to such labels could in principle learn the judge's idiosyncrasies rather than factual error, so we quantify how far the judge departs from human factuality judgments.

Annotating our own generations by hand at the required scale was not feasible, so we validated the judge on TruthfulQA (Lin et al., 2022), which supplies human reference labels for exactly this distinction. We drew $2{,}000$ samples and applied the short-answer QA labeling protocol, leaving $1{,}853$ after discarding responses initially judged hallucinated but judged factual on re-check. Human-annotated incorrect answers form the positive class. Table 15 reports the confusion matrix and the derived rates.

Table 15: GPT-5-mini agreement with TruthfulQA human labels ($n = 1{,}853$).

	Human: Incorrect	**Human: Correct**
Judge: Incorrect	869	83
Judge: Correct	4	897

Accuracy	**TPR**	**TNR**	**FPR**	**FNR**
95.30%	99.54%	91.53%	8.47%	0.46%

The judge agrees with human annotators on 95.3% of the retained TruthfulQA samples, and misses almost no human-identified error (FNR 0.46%). The residual disagreement is asymmetric: 8.5% of human-correct answers are marked incorrect by the judge, typically answers that are factually right but hedged or incompletely phrased. Our reported AUCs measure agreement with labels produced by the corresponding automatic protocol. The 95.3% agreement is specific to this TruthfulQA validation; it does not establish the accuracy of the LongFact labels, which rely solely on GPT-5 mini's internal knowledge without reference answers or external retrieval.

Discard statistics. The two-stage labeling protocol re-checks only responses initially judged hallucinated. A response is discarded only if the re-check judges it factual; it is not relabeled as factual. Initially factual responses and responses judged hallucinated in both passes are retained. On LongFact, both passes use only GPT-5 mini's internal knowledge because no reference answers are available. Table 16 reports QA discard rates; retained LongFact counts for each generation limit are in Table 12.

QA discard ratios are lowest on TriviaQA (2.6–3.4%), moderate on BioASQ (6.9–12.8%) and NQ (9.6–17.5%), and highest on SQuAD (17.8–24.7%). The SQuAD training-split rate grows from 17.8% (LLaMA-3.1-8B) to 21.2% (Mistral-Nemo-12B) to 23.9% (LLaMA-3.3-70B). All methods use the same filtered splits. LongFact discards 1.4–6.6%, below every QA benchmark except TriviaQA: with no reference answer both passes rely on the same internal knowledge, so they disagree less often than a reference-based first pass and a knowledge-based second pass do.

Table 16: Discard rate of automatically labeled samples (%). The second column covers the validation split as a whole, from which both validation and test are drawn.

Model	Dataset	Train	Validation + test
LLaMA-3.1-8B	TriviaQA	3.25%	2.85%
	SQuAD	17.75%	18.02%
	NQ	9.85%	9.62%
	BioASQ	6.91%	7.77%
	LongFact (max 256)	4.40%	5.40%
	LongFact (max 1024)	4.40%	4.88%
Mistral-Nemo-12B	TriviaQA	3.35%	3.20%
	SQuAD	21.15%	22.44%
	NQ	13.28%	13.44%
	BioASQ	10.70%	10.50%
	LongFact (max 256)	6.40%	5.56%
	LongFact (max 1024)	1.40%	2.40%
LLaMA-3.3-70B	TriviaQA	3.12%	2.64%
	SQuAD	23.85%	24.73%
	NQ	16.27%	17.51%
	BioASQ	12.78%	12.80%
	LongFact (max 256)	6.60%	5.76%
	LongFact (max 1024)	1.90%	1.64%

E Detailed Detection Results

E.1 Per-dataset throughput

Table 4 reports the arithmetic mean over the four QA benchmarks. Table 17 gives the corresponding per-dataset values.

Table 17: QA throughput (samples/s). LLMs Know uses the last token. Best and second-best are red and blue.

	LLaMA-3.1-8B				Mistral-Nemo-12B				LLaMA-3.3-70B			
Method	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ	TQ	SQ	NQ	BQ
SEP	7954	7679	7542	7668	7874	8072	7858	8016	7442	7358	7410	7414
LLMs Know	7662	7605	7369	7468	7172	7394	7375	7663	7100	7096	7066	7146
SAPLMA	4805	4758	4729	4718	4696	4712	4804	4779	4549	4485	4520	4678
HaMI	3612	3402	4096	4096	4025	3092	2886	2378	2349	2436	2315	2208
HaMI (SP)	0.17	0.09	0.10	0.17	0.28	0.12	0.22	0.13	0.14	0.10	0.18	0.19
Mean	5192	5274	5224	4649	5068	4237	3706	3657	2680	2637	2625	2402
Max	5446	5303	4992	5351	5202	4288	3720	3512	2707	2585	2602	2363

E.2 Per-dataset margins

Table 5 averages over the four QA benchmarks; the per-dataset values follow.

Table 18: Mean bag-wise margins by dataset. Red/blue: higher/lower within each block.

	LLaMA-3.1-8B				Mistral-Nemo-Instruct (12B)				LLaMA-3.3-Instruct-70B			
	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ
HaMI (Original)	6.17	2.09	3.16	4.28	5.97	3.66	3.42	4.33	5.12	3.67	3.56	4.06
HaMI (SP)	9.02	5.10	5.58	7.51	7.08	5.48	5.25	5.61	6.10	5.47	5.23	5.85
Mean Pooling	2.026	0.935	0.870	1.286	2.994	1.197	1.046	1.614	3.333	1.222	1.427	1.596
Max Pooling	2.299	1.073	1.052	1.498	3.404	1.420	1.231	1.924	3.722	1.366	1.640	1.824

E.3 Variability across Runs

Table 19 reports standard errors of detection AUC over five runs on the four QA benchmarks. Table 20 reports margin standard deviations over the same five runs.

Table 19: Test AUC standard errors over five runs on the four QA benchmarks.

	LLaMA-3.1-8B				Mistral-Nemo-Instruct (12B)				LLaMA-3.3-Instruct-70B			
	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ
SEP	0.003	0.002	0.002	0.005	0.004	0.004	0.004	0.005	0.005	0.001	0.002	0.006
LLMs Know (last token)	0.003	0.003	0.004	0.003	0.004	0.005	0.002	0.003	0.002	0.001	0.002	0.003
SAPLMA	0.005	0.003	0.002	0.002	0.002	0.003	0.003	0.003	0.002	0.001	0.002	0.003
HaMI (Original)	0.003	0.003	0.002	0.001	0.001	0.004	0.002	0.000	0.001	0.001	0.002	0.001
HaMI (SP)	0.001	0.000	0.001	0.001	0.001	0.002	0.001	0.001	0.002	0.003	0.001	0.001
Mean Pooling	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.001	0.000
Max Pooling	0.001	0.001	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.001	0.000

Table 20: Margin standard deviations over five runs.

	LLaMA-3.1-8B				Mistral-Nemo-Instruct (12B)				LLaMA-3.3-Instruct-70B			
	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ	TriviaQA	SQuAD	NQ	BioASQ
HaMI (Original)	0.128	0.354	0.120	0.247	0.215	0.270	0.104	0.420	0.126	0.181	0.116	0.344
HaMI (SP)	0.287	0.681	0.294	0.343	0.646	0.200	0.442	0.407	0.191	0.138	0.305	0.298
Mean Pooling	0.023	0.013	0.011	0.046	0.020	0.037	0.019	0.029	0.081	0.017	0.014	0.043
Max Pooling	0.014	0.031	0.056	0.028	0.062	0.026	0.022	0.028	0.095	0.021	0.025	0.021

Table 21: Test AUC standard errors over five runs on LongFact, at the two generation limits of Section 6.3.

	LLaMA-3.1-8B		Mistral-Nemo-12B		LLaMA-3.3-70B	
Method	256	1024	256	1024	256	1024
SEP	0.002	0.006	0.007	0.006	0.005	0.008
LLMs Know	0.008	0.008	0.011	0.015	0.006	0.018
SAPLMA	0.005	0.008	0.007	0.002	0.009	0.002
HaMI	0.024	0.029	0.003	0.001	0.025	0.011
HaMI (prob)	0.026	0.029	0.003	0.002	0.026	0.011
Mean	0.002	0.001	0.001	0.012	0.003	0.003
Max	0.003	0.001	0.003	0.002	0.002	0.007

Table 21 gives the same quantity for LongFact. The pooling detectors stay at or below 0.012 in every cell, while HaMI reaches 0.029 on LLaMA-3.1-8B and the last-token probe 0.018 on LLaMA-3.3-70B at 1024 tokens, the cell whose training split holds 35 hallucinated responses.

F Ablations and Sensitivity Analyses

F.1 Placement of Feature Extraction

To evaluate the effect of placing the feature extraction layer *before* pooling, we compared our architecture against a baseline that applied pooling directly to the raw hidden states ($\mathbb{R}^d$). In the latter case, the MLP was placed *after* the pooling operation. On the four QA benchmarks in Table 6, placing the feature extraction layer before pooling improves max-pooling AUROC by 0.7–5.7 points; mean-pooling AUROC changes by at most 0.2 points.

F.2 Impact of Feature Dimension D

Tables 23–25 report max-pooling AUC across feature dimensions. Performance generally improves beyond $D = 1$, with most gains between $D = 1$ and $D = 16$ and flatter curves around $D = 128$–256. This suggests that a moderate projection dimension can retain useful token-level information.

Mean vs. Max Pooling. Table 22 compares mean and max pooling on LLaMA-3.1-8B/TriviaQA. Both methods use the same dimension grid $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$, with all other hyperparameters held fixed. Both approach a plateau around $D = 256$, the dimension used in the main experiments. Max pooling achieves higher AUC at every tested dimension, with the largest gap at $D = 2$ (8.4 points), consistent with the dilution argument of Section 5.2.

Table 22: Mean- and max-pooling AUC by feature dimension (LLaMA-3.1-8B, TriviaQA), with shared training settings.

Dimension (D)	Mean Pooling	Max Pooling
1	0.770	0.833
2	0.782	0.866
4	0.851	0.885
8	0.871	0.912
16	0.881	0.917
32	0.887	0.920
64	0.892	0.922
128	0.895	0.923
256	0.896	0.924
512	0.897	0.925
1024	0.896	0.925

Table 23: Max-pooling AUC by feature dimension D: LLaMA-3.1-8B.

Dimension (D)	TriviaQA	SQuAD	NQ	BioASQ
1	0.833	0.775	0.777	0.826
2	0.866	0.793	0.791	0.845
4	0.885	0.806	0.799	0.863
8	0.912	0.814	0.808	0.886
16	0.917	0.822	0.813	0.894
32	0.920	0.823	0.814	0.895
64	0.922	0.824	0.816	0.897
128	0.923	0.826	0.816	0.899
256	0.924	0.827	0.815	0.899
512	0.925	0.828	0.815	0.900
1024	0.925	0.827	0.814	0.902

Table 24: Max-pooling AUC by feature dimension D: Mistral-Nemo-12B.

Dimension (D)	TriviaQA	SQuAD	NQ	BioASQ
1	0.917	0.833	0.821	0.874
2	0.920	0.839	0.836	0.876
4	0.930	0.847	0.842	0.888
8	0.932	0.847	0.842	0.890
16	0.936	0.850	0.845	0.893
32	0.937	0.854	0.848	0.896
64	0.939	0.856	0.849	0.897
128	0.940	0.857	0.850	0.897
256	0.940	0.855	0.850	0.896
512	0.938	0.854	0.849	0.894
1024	0.939	0.855	0.845	0.893

Table 25: Max-pooling AUC by feature dimension D: LLaMA-3.3-70B.

Dimension (D)	TriviaQA	SQuAD	NQ	BioASQ
1	0.691	0.746	0.827	0.772
2	0.789	0.770	0.832	0.811
4	0.899	0.852	0.880	0.898
8	0.936	0.855	0.881	0.908
16	0.940	0.862	0.886	0.912
32	0.939	0.867	0.887	0.914
64	0.940	0.870	0.888	0.914
128	0.943	0.871	0.889	0.915
256	0.944	0.871	0.888	0.916
512	0.944	0.872	0.889	0.915
1024	0.943	0.873	0.889	0.915

F.3 Layer-wise sensitivity

Layers are selected by validation AUC (Appendix A.4). Figures 4–6 show test AUC across the searched layers: 17 for LLaMA-3.1-8B, 21 for Mistral-Nemo-12B and 41 for LLaMA-3.3-70B, at a stride of two. Every point in this subsection is a mean of three runs with its standard error, not the five used for the reported numbers elsewhere.

On the short-answer benchmarks, the methods generally trace an inverted-U, peaking at roughly 40–45% of the network's depth (12–16 of 32, 16–20 of 40, 32–38 of 80); the peak location tracks relative rather than absolute depth, and it is set by the model and the task. The curves are flat near the peak: moving one stride step away from the selected layer costs a median of 0.5 AUC points for the pooling detectors, and stays under 1.3 points in nine cases out of ten. The LongFact panels show the 256-token setting of the response-length study; Table 14 gives its validation-selected layers.

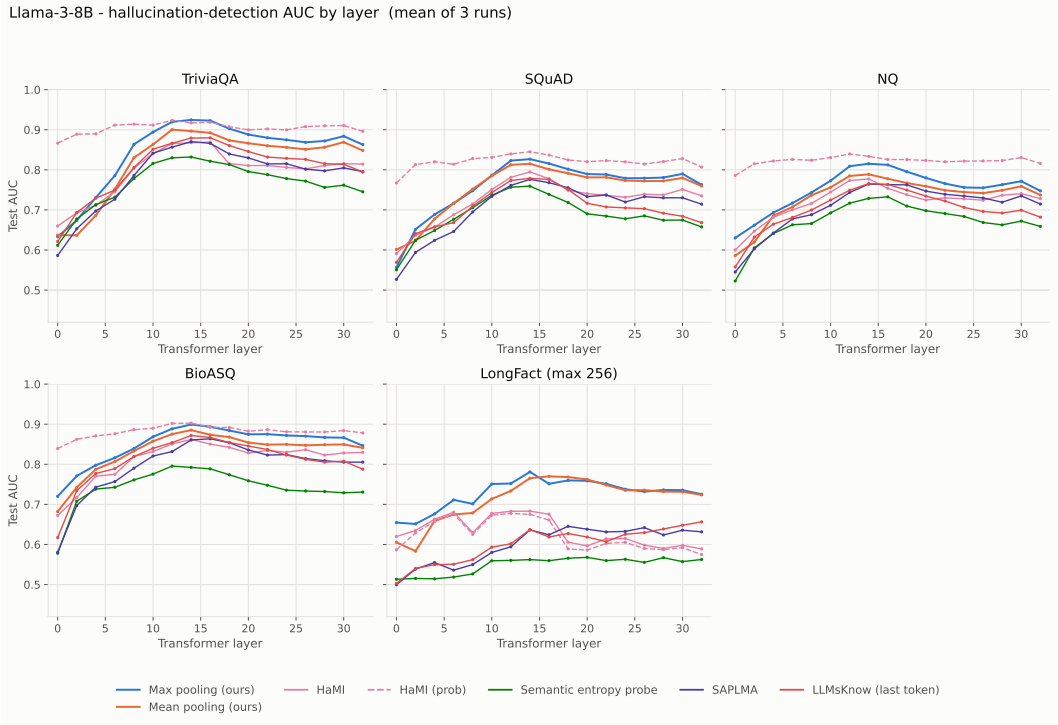


Figure 4: LLaMA-3.1-8B layer sensitivity (17 layers, stride two). Curves show mean test AUC over three runs; the LongFact panel is the 256-token setting of Section 6.3.

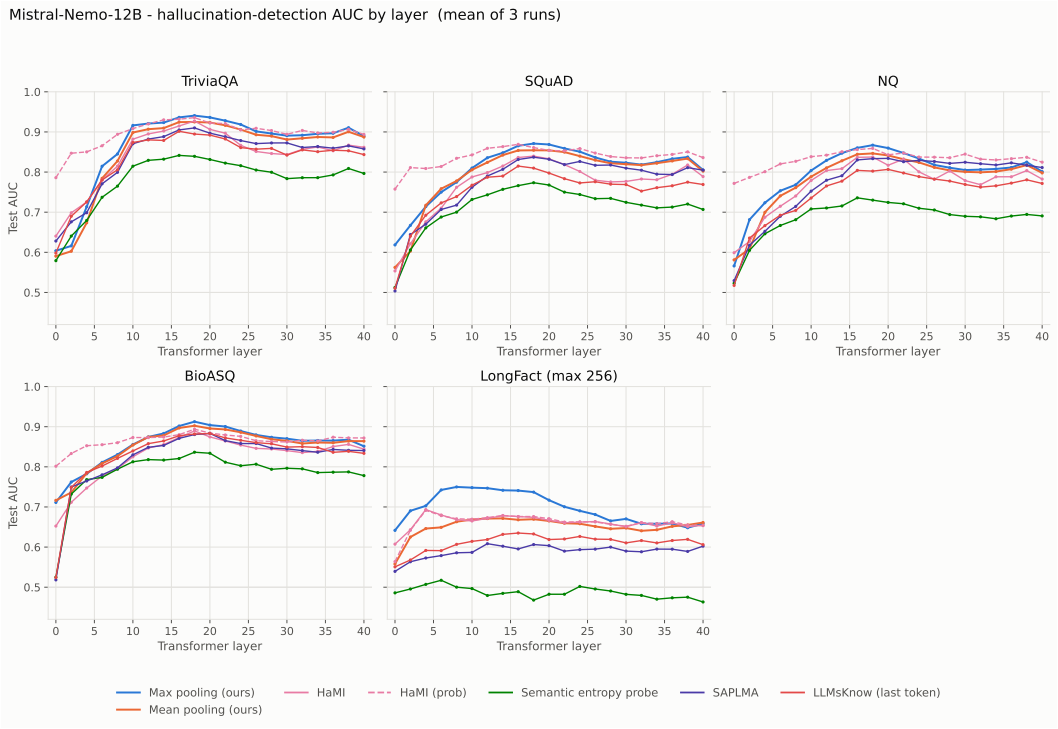


Figure 5: Mistral-Nemo-12B layer sensitivity (21 layers, stride two). Conventions as in Figure 4.

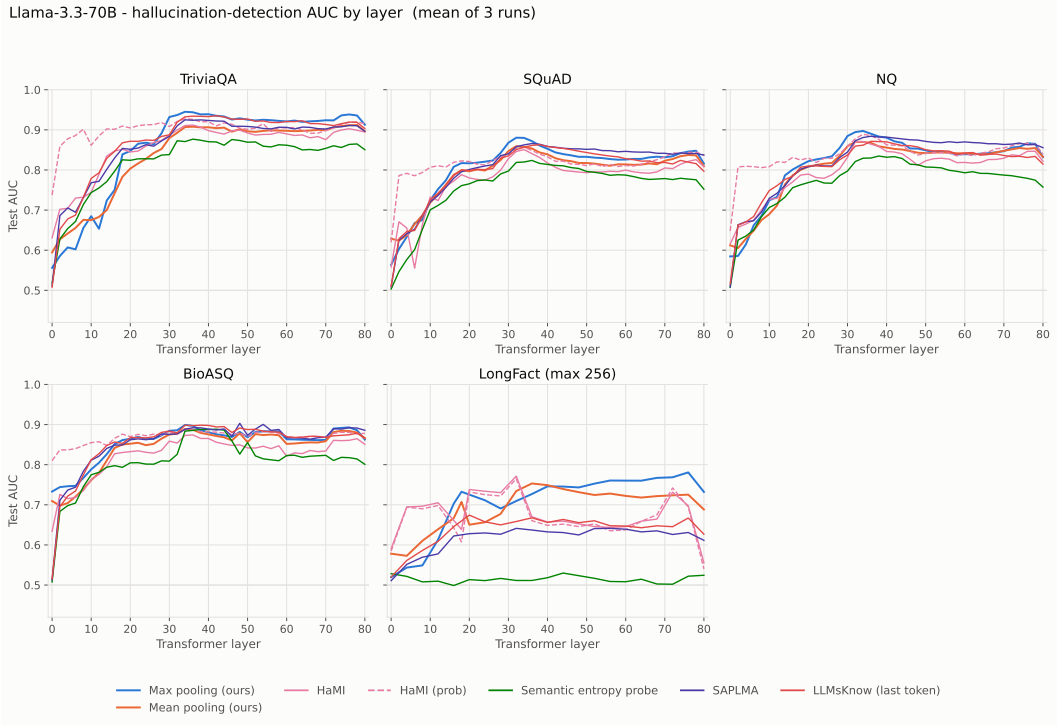


Figure 6: LLaMA-3.3-70B layer sensitivity (41 layers, stride two). Conventions as in Figure 4.

G Empirical Observations in HaMI

G.1 Summary of Expected Values

Tables 26–31 summarize semantic probability, local sensitivity, and their joint product for each class. The joint product determines γ in Equation (4); the required quantity is the negative-to-positive class ratio, not the negative-class fraction. Each table is computed on the bags its own row reports, which is a subset of the split counted in Table 10; the condition is checked against the ratio of the same subset. Table 10 supplies the class ratios for all three generators, so the condition can be checked directly: on every model–dataset pair among the four short-answer benchmarks the measured ratio exceeds the corresponding $1/\gamma$ threshold, the tightest case being LLaMA-3.1-8B on SQuAD (0.52 against 0.39). Appendix H repeats the analysis with integrated sensitivities.

Table 26: Empirical expected values and sensitivity ratios for **LLaMA-3.1-8B** (**Train**).

	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.430	**0.914**	13.4	**18.8**	6.49	**17.5**	**2.70**	**0.37**
SQuAD	0.352	**0.730**	13.4	**18.6**	5.42	**14.8**	**2.73**	**0.37**
NQ	0.382	**0.772**	13.0	**16.8**	5.52	**13.7**	**2.48**	**0.40**
BioASQ	0.392	**0.756**	12.0	**16.0**	5.06	**12.8**	**2.53**	**0.40**

Table 27: Empirical expected values and sensitivity ratios for **LLaMA-3.1-8B** (**Test**).

	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.451	**0.926**	13.8	**18.9**	7.09	**17.8**	**2.51**	**0.40**
SQuAD	0.360	**0.733**	13.5	**18.5**	5.67	**14.6**	**2.57**	**0.39**
NQ	0.383	**0.732**	12.9	**16.4**	5.53	**12.8**	**2.31**	**0.43**
BioASQ	0.380	**0.764**	12.1	**16.1**	4.98	**13.0**	**2.61**	**0.38**

Table 28: Empirical expected values and sensitivity ratios for **Mistral-12B** (**Train**).

	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.581	**0.960**	15.1	**20.8**	9.40	**20.2**	**2.15**	**0.47**
SQuAD	0.457	**0.835**	9.60	**15.1**	5.09	**13.3**	**2.61**	**0.38**
NQ	0.535	**0.886**	10.5	**15.0**	6.43	**13.8**	**2.15**	**0.47**
BioASQ	0.480	**0.881**	8.54	**11.7**	4.53	**10.6**	**2.34**	**0.43**

Table 29: Empirical expected values and sensitivity ratios for **Mistral-12B** (**Test**).

	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.590	**0.966**	15.6	**20.8**	10.0	**20.3**	**2.03**	**0.49**
SQuAD	0.502	**0.849**	10.4	**15.0**	6.04	**13.3**	**2.20**	**0.45**
NQ	0.525	**0.881**	10.6	**14.8**	6.50	**13.5**	**2.08**	**0.48**
BioASQ	0.491	**0.898**	8.48	**11.7**	4.65	**10.8**	**2.32**	**0.43**

Table 30: Empirical expected values and sensitivity ratios for **LLaMA-3.3-70B** (**Train**).

	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.484	**0.944**	10.0	**13.8**	5.43	**13.2**	**2.43**	**0.41**
SQuAD	0.439	**0.842**	8.66	**10.8**	4.14	**9.40**	**2.27**	**0.44**
NQ	0.538	**0.888**	8.42	**11.1**	4.97	**10.1**	**2.03**	**0.49**
BioASQ	0.459	**0.866**	9.12	**11.9**	4.49	**10.6**	**2.36**	**0.42**

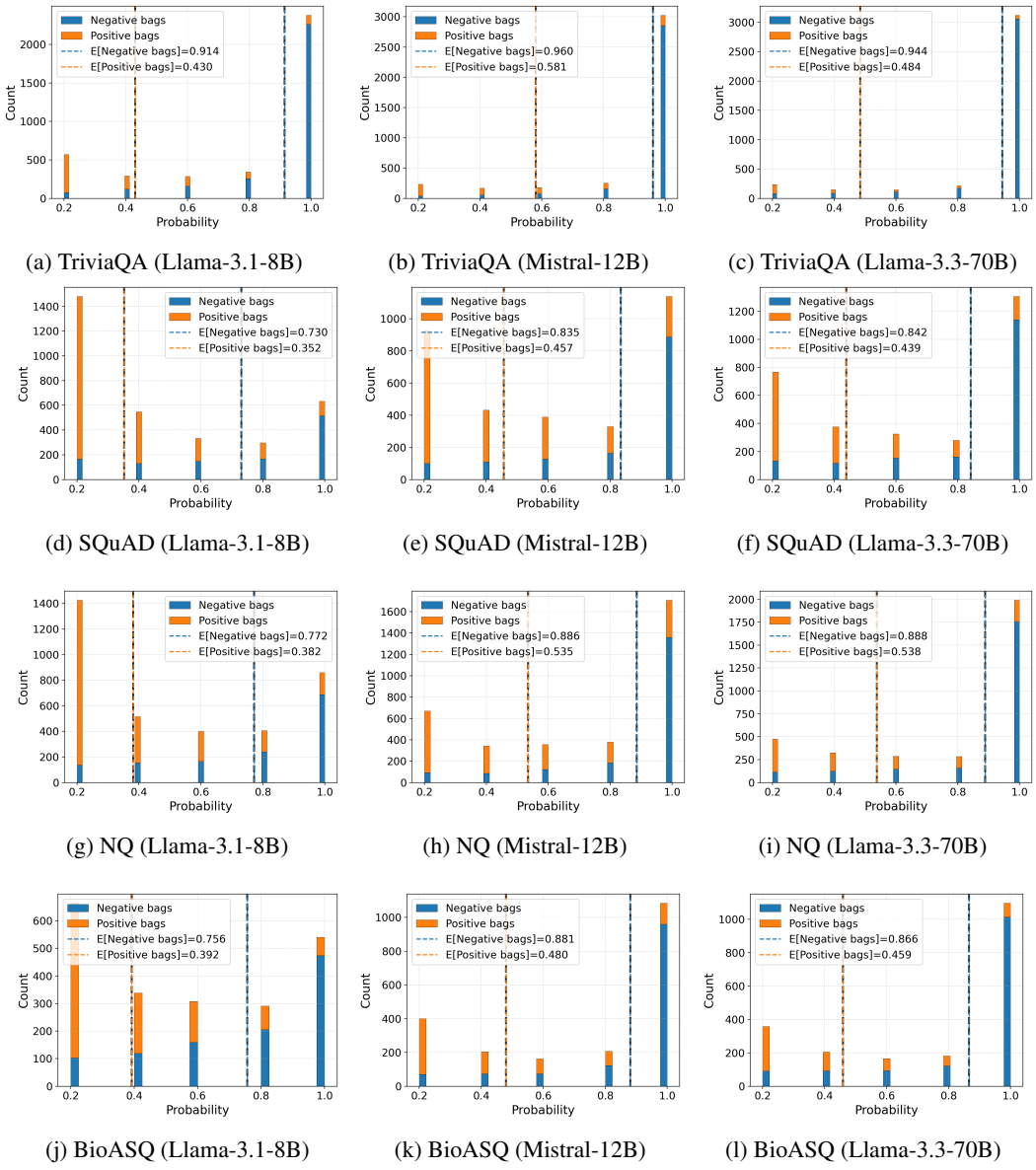
Table 31: Empirical expected values and sensitivity ratios for **LLaMA-3.3-70B** (**Test**).

	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	Pos ($\mathbb{E}_{pos}$)	Neg ($\mathbb{E}_{neg}$)	γ	$1/\gamma$
TriviaQA	0.554	**0.940**	10.6	**13.7**	6.44	**13.1**	**2.03**	**0.49**
SQuAD	0.471	**0.843**	8.79	**10.8**	4.44	**9.34**	**2.10**	**0.48**
NQ	0.525	**0.894**	8.27	**11.0**	4.81	**10.0**	**2.08**	**0.48**
BioASQ	0.490	**0.866**	9.16	**11.8**	4.85	**10.5**	**2.16**	**0.46**

G.2 Distribution of Semantic Probability

Figures 7 (train) and 8 (test) show the empirical distribution of semantic probabilities $P_{\text{sem}}^{\mathbf{B}}$. Across all datasets and models, negative bags exhibit a higher density in high-probability regions than positive bags, confirming the relationship $\mathbb{E}_{neg}[P_{\text{sem}}^{\mathbf{B}}] > \mathbb{E}_{pos}[P_{\text{sem}}^{\mathbf{B}}]$. The expected values for each condition

are provided in Appendix G.1. The AUC performance is summarized in Table 32. $P_{\text{sem}}^{\mathbf{B}}$ consistently achieves high discriminative power.



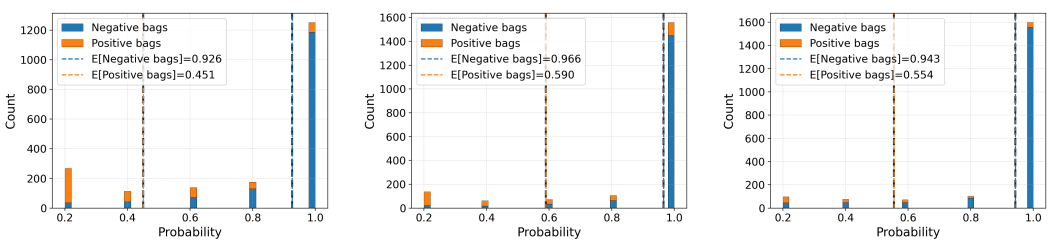
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)

(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)

(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)

(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

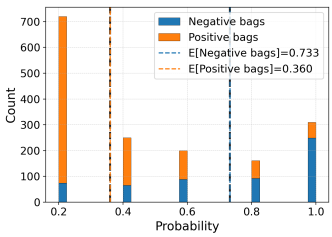
Figure 7: Empirical distribution of semantic probabilities $P_{\text{sem}}^{\mathbf{B}}$ in **train** data.



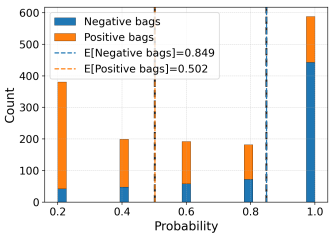
(a) TriviaQA (Llama-3.1-8B)

(b) TriviaQA (Mistral-12B)

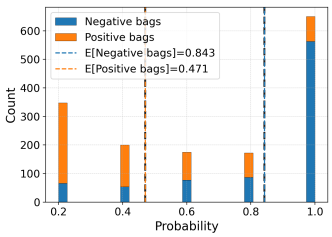
(c) TriviaQA (Llama-3.3-70B)



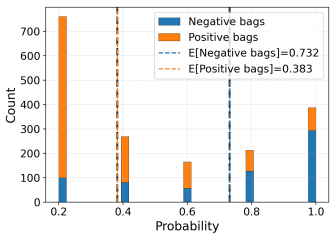
(d) SQuAD (Llama-3.1-8B)



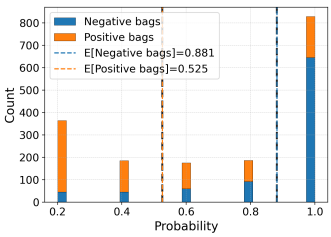
(e) SQuAD (Mistral-12B)



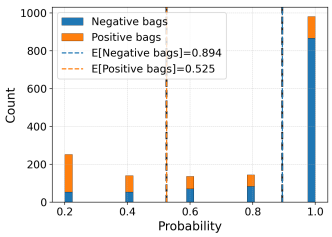
(f) SQuAD (Llama-3.3-70B)



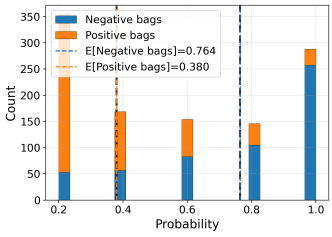
(g) NQ (Llama-3.1-8B)



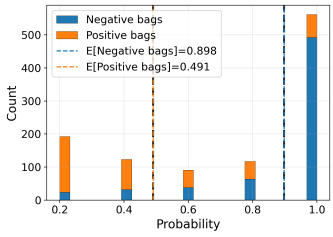
(h) NQ (Mistral-12B)



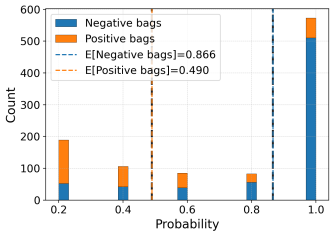
(i) NQ (Llama-3.3-70B)



(j) BioASQ (Llama-3.1-8B)



(k) BioASQ (Mistral-12B)



(l) BioASQ (Llama-3.3-70B)

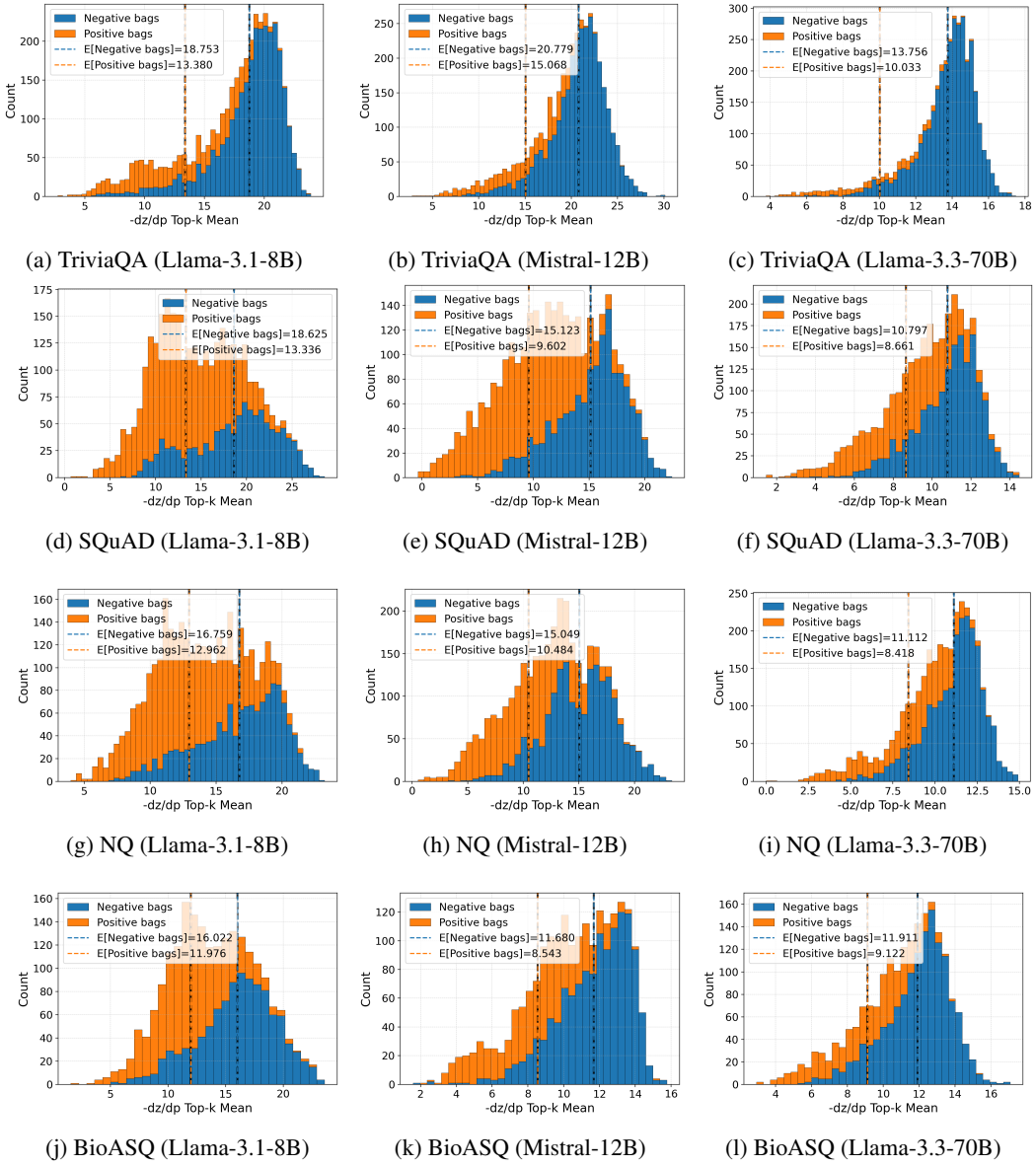
Figure 8: Empirical distribution of semantic probabilities $P_{\text{sem}}^{\mathbf{B}}$ in **test** data.

Table 32: AUC scores for Semantic Probability as a discriminator across datasets and models.

Dataset	**LLaMA-3.1-8B**	**Mistral-12B**	**LLaMA-3.3-70B**
TriviaQA	0.883	0.812	0.832
SQuAD	0.812	0.802	0.817
NQ	0.800	0.794	0.807
BioASQ	0.840	0.843	0.810

G.3 DISTRIBUTION OF $\bar{C}_B$

To examine $\mathbb{E}_{neg}[\bar{C}_{\mathbf{B}}] > \mathbb{E}_{pos}[\bar{C}_{\mathbf{B}}]$, we analyzed the sensitivity distributions in Figures 9–10 and the class-wise means in Appendix G.1. Across the evaluated models and four short-answer benchmarks, the distributions for negative bags tend to be shifted toward larger values. The higher empirical means indicate that the trained network exhibits greater sensitivity to input scaling for negative bags on average. This pattern is observed in both data splits.



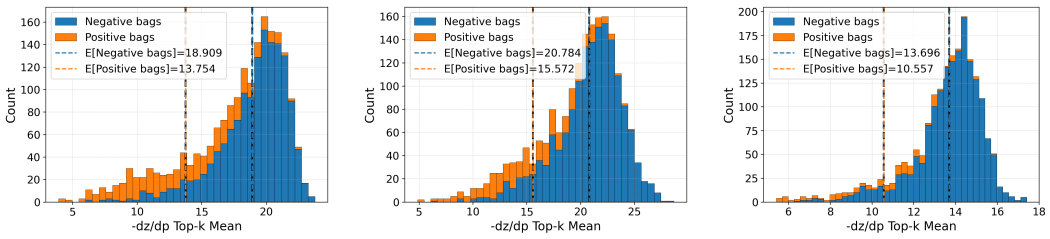
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)

(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)

(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)

(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

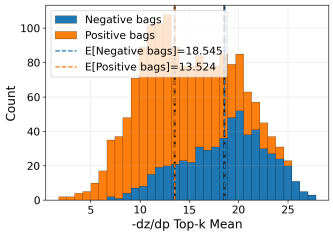
Figure 9: Empirical distribution of the bag-level sensitivity to input scaling $\bar{C}_{\mathbf{B}}$ in **train** data.



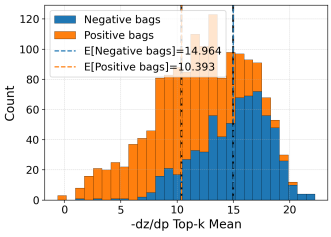
(a) TriviaQA (Llama-3.1-8B)

(b) TriviaQA (Mistral-12B)

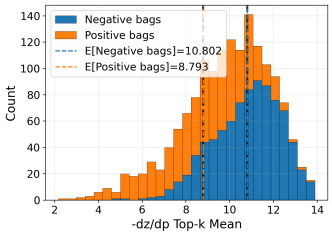
(c) TriviaQA (Llama-3.3-70B)



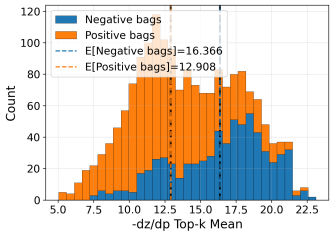
(d) SQuAD (Llama-3.1-8B)



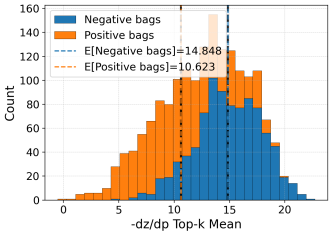
(e) SQuAD (Mistral-12B)



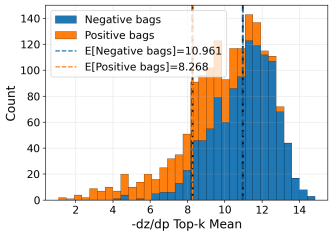
(f) SQuAD (Llama-3.3-70B)



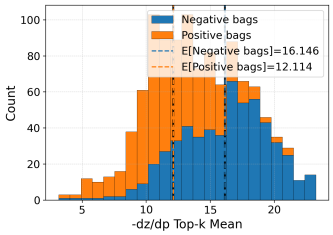
(g) NQ (Llama-3.1-8B)



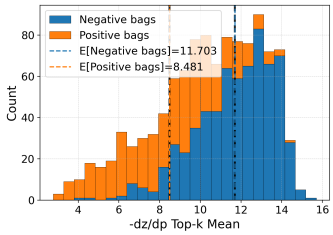
(h) NQ (Mistral-12B)



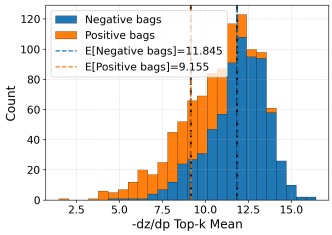
(i) NQ (Llama-3.3-70B)



(j) BioASQ (Llama-3.1-8B)



(k) BioASQ (Mistral-12B)

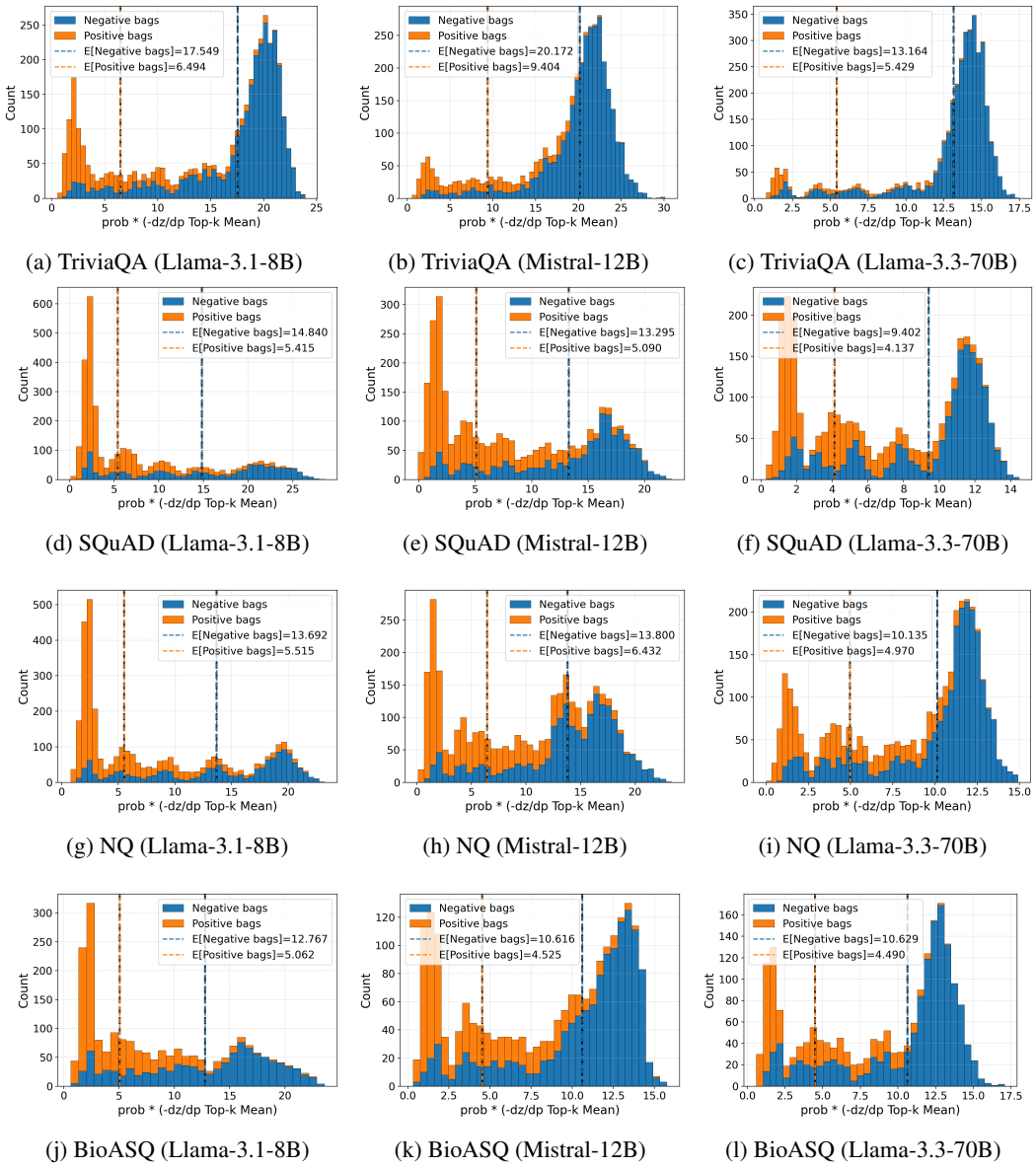


(l) BioASQ (Llama-3.3-70B)

Figure 10: Empirical distribution of the bag-level sensitivity to input scaling $\bar{C}_{\mathbf{B}}$ in **test** data.

G.4 DISTRIBUTION OF $P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}$

We computed the joint product $P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}$ for each bag. Figures 11 and 12 illustrate the empirical distributions of this product for the training and test sets, respectively. Although the distributions of individual bags naturally exhibit some overlap, the negative class consistently demonstrates a higher density at larger values. We therefore summarize the expected values for each class. Although there is an overlap in the distributions of individual bags, the class-wise expected values satisfy the inequality $\mathbb{E}_{neg}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}] > \mathbb{E}_{pos}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}]$ across all evaluated datasets and across both data splits. The expected values for each condition are provided in Appendix G.1.



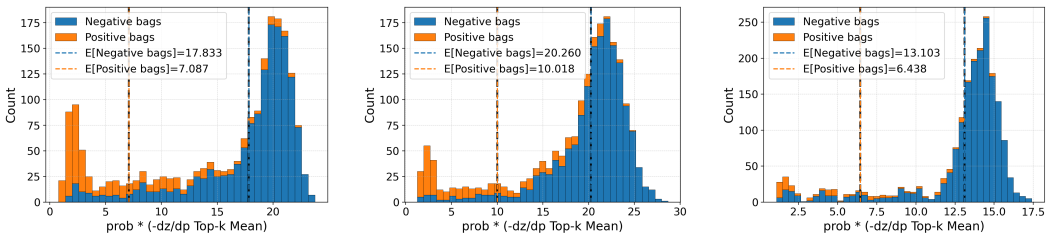
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)

(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)

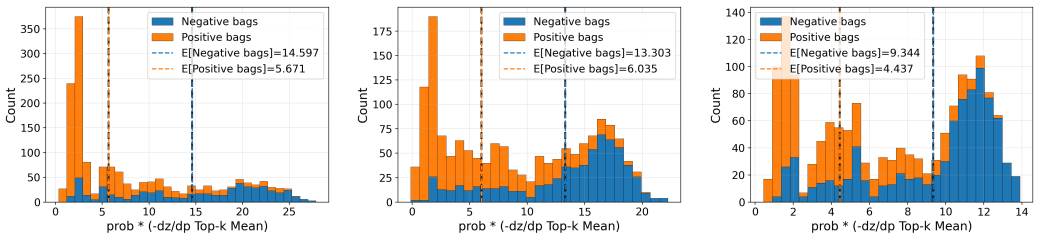
(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)

(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

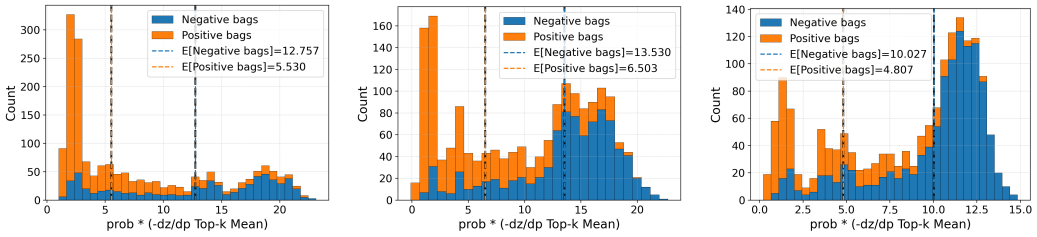
Figure 11: Distribution of the joint product $P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}$ in **train** data.



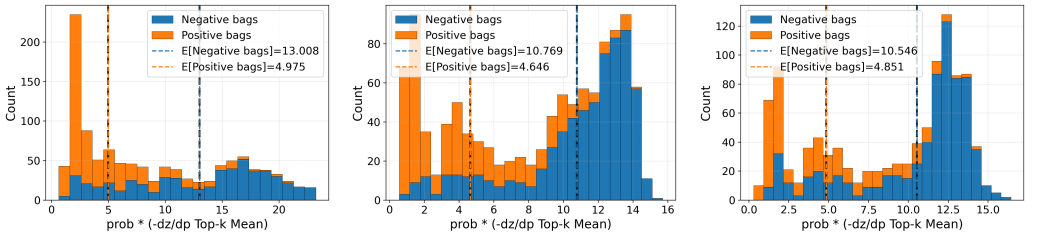
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)



(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)



(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)



(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

Figure 12: Distribution of the joint product $P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}$ in **test** data.

H Empirical Checks with Integrated Sensitivity

H.1 Summary of Expected Values via Path Integration

These tables use $\lambda = 1$ and $N = 1000$ integration steps to estimate $\bar{C}^{\text{int}}_{\mathbf{B}}$, as defined in Appendix C. As in Appendix G.1, the negative class has the larger joint expectation. The margin condition is satisfied for all three generators on the four short-answer benchmarks, using the class ratios of Table 10.

Table 33: Empirical expected values for **LLaMA-3.1-8B** (**Train**) under non-invariant conditions.

	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.430	**0.914**	13.2	**19.0**	6.59	**17.8**	**2.70**	**0.37**
SQuAD	0.352	**0.730**	12.6	**18.8**	5.27	**15.0**	**2.85**	**0.35**
NQ	0.382	**0.772**	12.8	**16.9**	5.59	**13.8**	**2.47**	**0.41**
BioASQ	0.392	**0.756**	11.9	**16.3**	5.11	**13.0**	**2.54**	**0.39**

Table 34: Empirical expected values for **LLaMA-3.1-8B** (**Test**) under non-invariant conditions.

	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.451	**0.926**	13.7	**19.2**	7.27	**18.1**	**2.49**	**0.40**
SQuAD	0.360	**0.733**	13.1	**18.8**	5.67	**14.9**	**2.63**	**0.38**
NQ	0.383	**0.732**	12.8	**16.6**	5.63	**13.0**	**2.31**	**0.43**
BioASQ	0.380	**0.764**	12.2	**16.6**	5.10	**13.3**	**2.61**	**0.38**

Table 35: Empirical expected values for **Mistral-12B** (**Train**) under non-invariant conditions.

	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.581	**0.960**	15.0	**20.6**	9.45	**20.0**	**2.12**	**0.47**
SQuAD	0.457	**0.835**	9.40	**15.1**	5.08	**13.3**	**2.62**	**0.38**
NQ	0.535	**0.886**	10.3	**15.2**	6.44	**13.9**	**2.16**	**0.46**
BioASQ	0.480	**0.881**	8.55	**11.7**	4.68	**10.6**	**2.26**	**0.44**

Table 36: Empirical expected values for **Mistral-12B** (**Test**) under non-invariant conditions.

	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.590	**0.966**	15.8	**20.8**	10.4	**20.2**	**1.94**	**0.51**
SQuAD	0.502	**0.849**	10.4	**15.1**	6.12	**13.4**	**2.19**	**0.46**
NQ	0.525	**0.881**	10.5	**15.0**	6.58	**13.7**	**2.08**	**0.48**
BioASQ	0.491	**0.898**	8.63	**11.9**	4.85	**10.9**	**2.25**	**0.44**

Table 37: Empirical expected values for **LLaMA-3.3-70B** (**Train**) under non-invariant conditions.

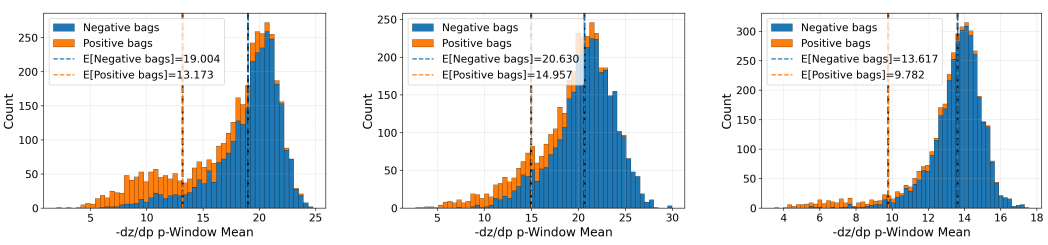
	Semantic Prob. $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}]$		**Sensitivity** $\mathbb{E}[\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Joint Product** $\mathbb{E}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.484	**0.944**	9.78	**13.6**	5.50	**13.0**	**2.36**	**0.42**
SQuAD	0.439	**0.842**	8.63	**10.8**	4.29	**9.35**	**2.18**	**0.46**
NQ	0.538	**0.888**	8.64	**11.1**	5.27	**10.1**	**1.92**	**0.52**
BioASQ	0.459	**0.866**	9.04	**11.9**	4.59	**10.6**	**2.31**	**0.43**

Table 38: Empirical expected values for **LLaMA-3.3-70B** (**Test**) under non-invariant conditions.

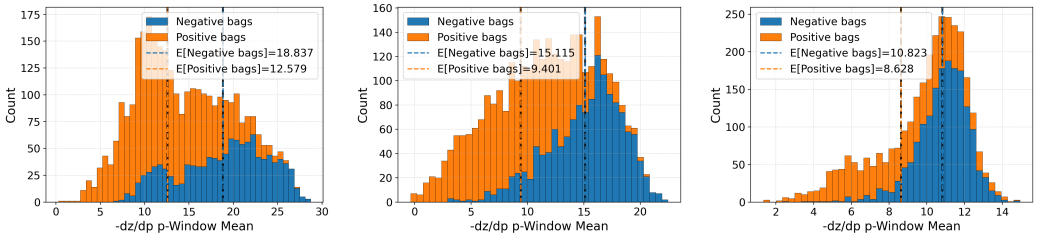
	Semantic Prob. $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}]$		**Sensitivity** $\mathbb{E}[\bar{C}_{\mathbf{B}}^{\text{int}}]$		**Joint Product** $\mathbb{E}[P_{\text{sem}}^{\mathbf{B}}\bar{C}_{\mathbf{B}}^{\text{int}}]$		**Ratios**	
Dataset	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	Pos ($\mathbb{E}_{\text{pos}}$)	Neg ($\mathbb{E}_{\text{neg}}$)	γ	$1/\gamma$
TriviaQA	0.554	**0.940**	10.6	**13.6**	6.65	**13.0**	**1.95**	**0.51**
SQuAD	0.471	**0.843**	9.01	**10.9**	4.70	**9.42**	**2.00**	**0.50**
NQ	0.525	**0.894**	8.52	**11.1**	5.08	**10.1**	**1.99**	**0.50**
BioASQ	0.490	**0.866**	9.45	**12.0**	5.17	**10.6**	**2.05**	**0.49**

H.2 Distribution of Integrated Sensitivity $\bar{C}_{\mathbf{B}}^{\text{int}}$

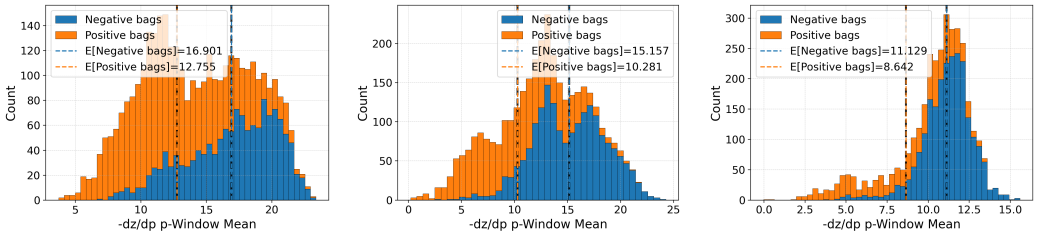
We analyzed the sensitivity distributions in Figures 13–14. All values were calculated with $\lambda = 1$ and using the integrated cost $\bar{C}_{\mathbf{B}}^{\text{int}}$ via numerical approximation with $N = 1000$ steps. Across the evaluated models and four short-answer benchmarks, the distributions for negative bags tend to be shifted toward larger values. The class-wise means in Section H.1 show greater integrated sensitivity for negative bags on average in both data splits.



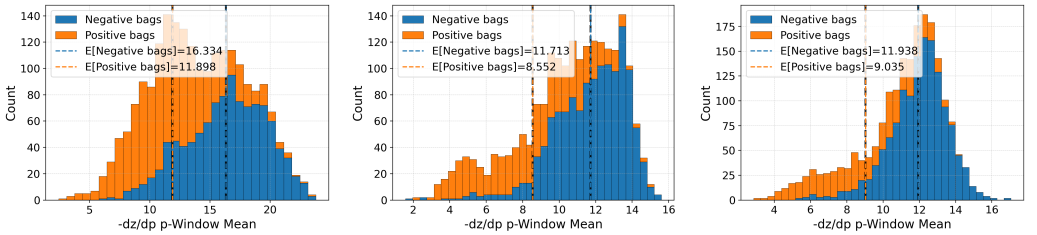
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)



(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)

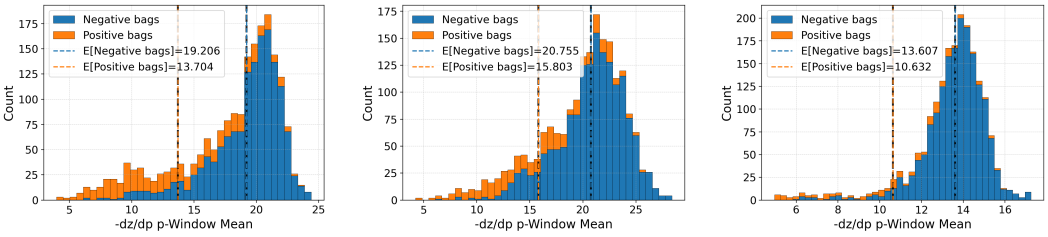


(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)

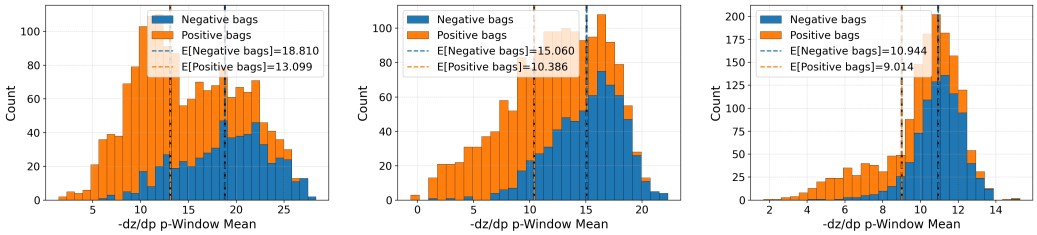


(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

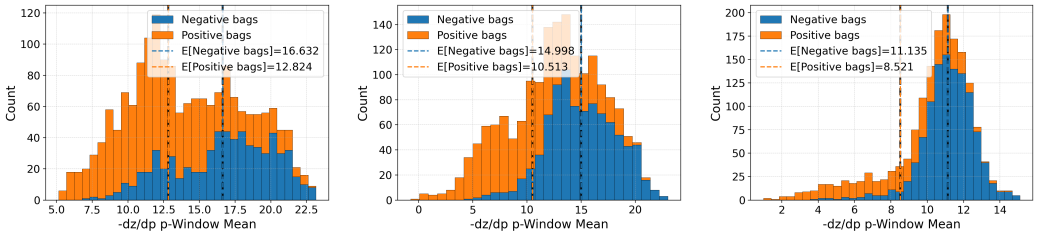
Figure 13: Empirical distribution of $\bar{C}_{\mathbf{B}}^{\text{int}}$ in **train** data (Non-invariant case).



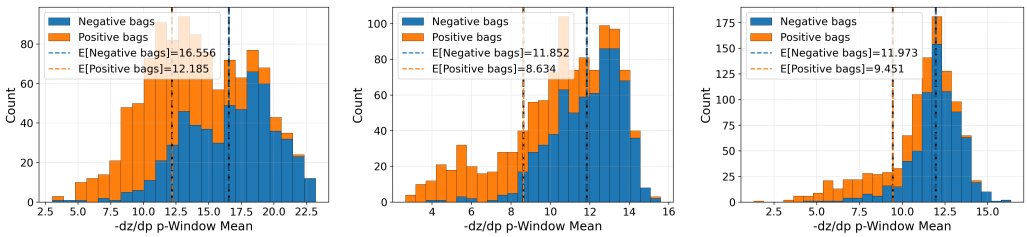
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)



(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)



(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)



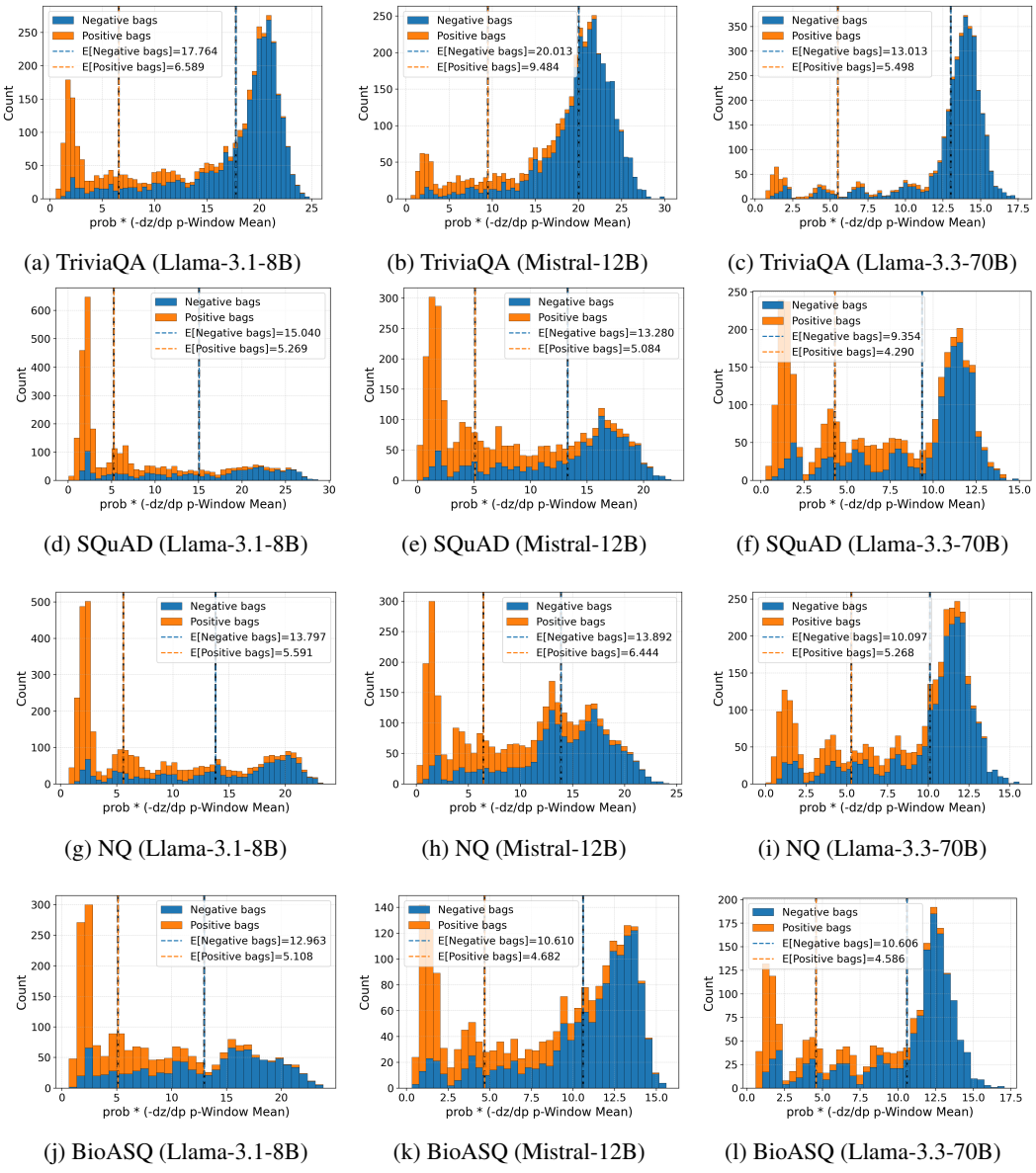
(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

Figure 14: Empirical distribution of $\bar{C}_{\mathbf{B}}^{\text{int}}$ in **test** data (Non-invariant case).

H.3 DISTRIBUTION OF $P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}$

We computed the joint product $P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}$ for each bag (Figures 15–16). All values were calculated with $\lambda = 1$ and using the integrated cost $\bar{C}^{\text{int}}_{\mathbf{B}}$ via numerical approximation with $N = 1000$ steps.

Although individual bag distributions exhibit some overlap, the negative class demonstrates a higher density at larger values. The empirical class-wise expectations satisfy $\mathbb{E}_{\text{neg}}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}] > \mathbb{E}_{\text{pos}}[P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}]$ across all evaluated datasets and splits. The expected values for each condition are provided in Section H.1.



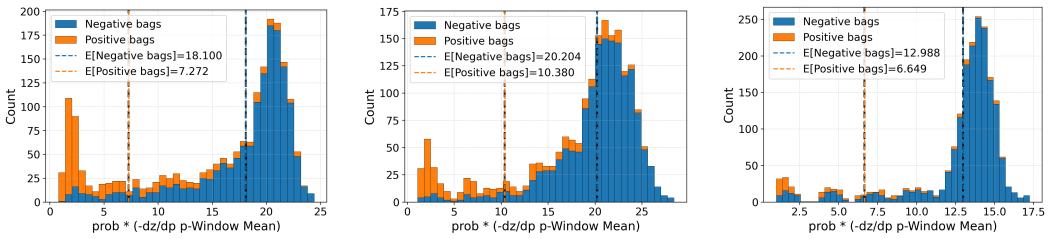
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)

(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)

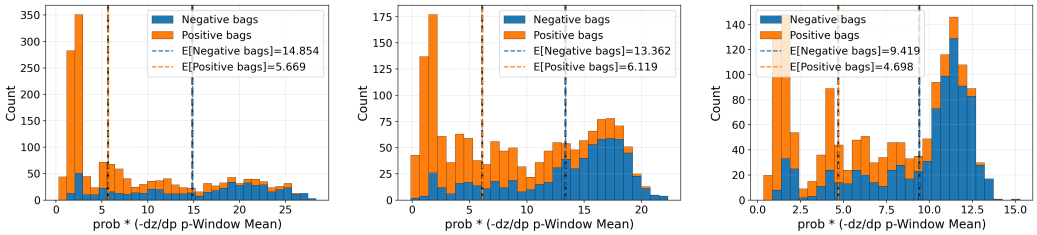
(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)

(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

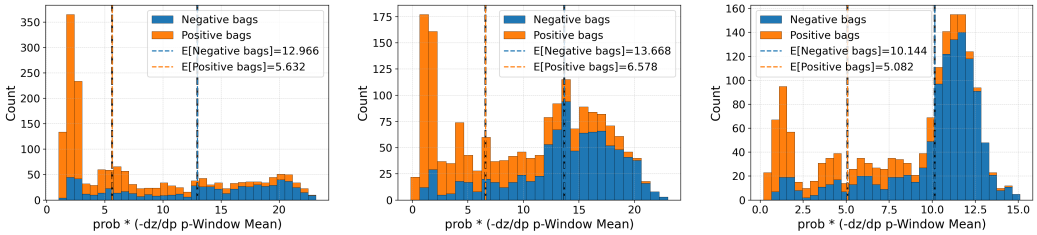
Figure 15: Distribution of the joint product $P^{\mathbf{B}}_{\text{sem}}\bar{C}^{\text{int}}_{\mathbf{B}}$ in **train** data.



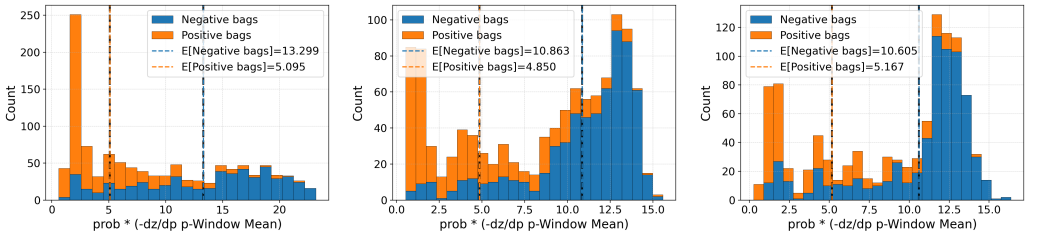
(a) TriviaQA (Llama-3.1-8B) (b) TriviaQA (Mistral-12B) (c) TriviaQA (Llama-3.3-70B)



(d) SQuAD (Llama-3.1-8B) (e) SQuAD (Mistral-12B) (f) SQuAD (Llama-3.3-70B)



(g) NQ (Llama-3.1-8B) (h) NQ (Mistral-12B) (i) NQ (Llama-3.3-70B)



(j) BioASQ (Llama-3.1-8B) (k) BioASQ (Mistral-12B) (l) BioASQ (Llama-3.3-70B)

Figure 16: Distribution of the joint product $P_{\text{sem}}^{\mathbf{B}} \bar{C}_{\mathbf{B}}^{\text{int}}$ in **test** data.