

---

# Detecting AI Hallucinations in Finance: An Information-Theoretic Method Cuts Hallucination Rate by 92%

---

**Mainak Singha**

Astrophysics Science Division, NASA, Goddard Space Flight Center,  
8800 Greenbelt Road, MD 20771

Department of Physics, The Catholic University of America, Washington, DC 20064

`mainak.singha@nasa.gov, singham@cua.edu`

## Abstract

Large language models (LLMs) produce fluent but unsupported answers—*hallucinations*—limiting safe deployment in high-stakes domains. We propose ECLIPSE, a framework that treats hallucination as a mismatch between a model’s *semantic entropy* and the *capacity* of available evidence. We combine entropy estimation via multi-sample clustering with a novel *perplexity decomposition* that measures how models use retrieved evidence. We prove that under mild conditions, the resulting entropy–capacity objective is strictly convex with a unique stable optimum. We evaluate on a controlled financial question answering dataset with GPT-3.5-turbo ( $n = 200$  balanced samples with synthetic hallucinations), where ECLIPSE achieves ROC AUC of 0.89 and average precision of 0.90, substantially outperforming a semantic entropy-only baseline (AUC 0.50). A controlled ablation with Claude-3-Haiku, which lacks token-level log probabilities, shows AUC dropping to 0.59 with coefficient magnitudes decreasing by 95%—demonstrating that ECLIPSE is a *logprob-native* mechanism whose effectiveness depends on calibrated token-level uncertainties. The perplexity decomposition features exhibit the largest learned coefficients, confirming that evidence utilization is central to hallucination detection. We position this work as a controlled mechanism study; broader validation across domains and naturally occurring hallucinations remains future work.

## 1 Introduction

Large language models have become integral to decision support in finance [27], healthcare [25], and law [2]. However, their tendency to produce confident but factually incorrect statements—commonly termed *hallucinations*—poses significant risks in these high-stakes domains [12, 10]. Hallucinations are particularly dangerous because models often express them with high apparent certainty, making them difficult to detect through simple confidence thresholds [28].

Existing hallucination detection methods primarily measure model uncertainty through semantic entropy [15] or self-consistency checks [19]. However, uncertainty alone provides an incomplete picture: a model may exhibit honest uncertainty when evidence is weak, or dangerous overconfidence when evidence is strong but ignored. We argue that hallucination risk depends fundamentally on the *relationship* between two quantities: the model’s expressed certainty and the quality of available evidence.

We propose ECLIPSE (Entropy–Capacity Logprob-Native Inference for Predicting Spurious Emissions), a framework that makes this entropy–capacity trade-off explicit. We call a method *logprob-native* if its core signal relies directly on token-level log probabilities and degrades substantially when those probabilities are replaced by uninformative proxies. Unlike semantic entropy alone, which cannot distinguish honest uncertainty from evidence-ignoring behavior, ECLIPSE decomposes perplexity to measure *how* models use evidence. Unlike Semantic Entropy Probes (SEPs), which require white-box access to hidden states, ECLIPSE achieves comparable detection performance using only API-accessible log probabilities. Unlike SelfCheckGPT, which relies on self-consistency across generations, ECLIPSE directly measures the evidence–answer relationship through perplexity decomposition.

Our contributions are:

1. **Theoretical framework:** We introduce a joint objective over semantic entropy  $H$  and evidence capacity  $C$  and prove it is strictly convex under mild conditions (Theorem 4), providing a principled foundation for hallucination risk modeling and, in future work, control.
2. **Perplexity decomposition:** We extract features  $(L_Q, L_{QE}, \Delta L)$  that decompose *how* models use evidence, enabling fine-grained detection of evidence-ignoring behavior.
3. **Empirical validation:** On our controlled financial QA dataset ( $n = 200$ ), ECLIPSE achieves 0.89 AUC using only API access (Figure 1), indicating that logprob-native, grey-box detectors can reach strong performance without hidden-state access.
4. **Mechanism validation:** We show through controlled ablation that ECLIPSE is logprob-native: without real log probabilities, performance drops from 0.89 to 0.59 AUC, only modestly above chance (Figure 5).

## 2 Related Work

**Hallucination in language models.** The phenomenon of LLM hallucination has received substantial attention [12, 31, 10]. Hallucinations manifest in various forms: factual errors [21], entity confusion [3], and fabricated citations [18]. Maynez et al. [20] categorize hallucinations in abstractive summarization as intrinsic (contradicting source) or extrinsic (unverifiable from source). Our work focuses on the latter in retrieval-augmented settings.

**Uncertainty quantification in LLMs.** Gal and Ghahramani [5] established dropout-based uncertainty estimation for neural networks. For LLMs, Kadavath et al. [13] demonstrated that models can partially assess their own uncertainty but degrade out-of-distribution. Kuhn et al. [15] introduced semantic entropy, which clusters sampled outputs by meaning and computes entropy over clusters. Lin et al. [17] extended this with conformal prediction for calibrated uncertainty sets. We build on semantic entropy but augment it with evidence capacity to distinguish honest uncertainty from hallucination.

**Hallucination detection methods.** SelfCheckGPT [19] samples multiple responses and measures consistency, achieving strong results on biography generation. Semantic Entropy Probes (SEPs) [14] train linear classifiers on hidden states, reaching 0.85–0.90 AUC but requiring white-box model access. SAPLMA [1] similarly uses hidden state classifiers. Min et al. [21] decompose responses into atomic facts and verify each against knowledge sources. Unlike these approaches, ECLIPSE achieves comparable performance using only API access while providing interpretable coefficients.

**Retrieval-augmented generation.** RAG systems [16, 8, 11] ground model outputs in retrieved documents but do not eliminate hallucinations [24]. Shi et al. [23] show that irrelevant context can degrade performance. Yoran et al. [30] propose filtering retrieved passages by relevance. Our perplexity decomposition specifically targets the failure mode where models ignore relevant evidence.

**Calibration and selective prediction.** Guo et al. [7] demonstrated that modern neural networks are poorly calibrated. Temperature scaling [7] and Platt scaling [22] provide post-hoc calibration. Selective prediction frameworks [4, 6] enable models to abstain when uncertain. Varshney et al. [26]

apply selective prediction to question answering. ECLIPSE provides a principled abstention signal based on both uncertainty and evidence quality.

**Information-theoretic approaches.** Xu et al. [29] analyze neural text generation through the lens of information theory. Holtzman et al. [9] identify the ‘‘likelihood trap’’ where high-probability text is often degenerate. Our capacity measure draws on information-theoretic intuitions, quantifying mutual information between evidence and answer through log-likelihood differences.

### 3 Method: ECLIPSE

#### 3.1 Problem Setup

We consider a language model  $p_\theta(A | Q, E)$  that answers query  $Q$  given evidence  $E$ , producing answer  $A$ . We seek to estimate the probability that  $A$  is hallucinated—unsupported or contradicted by  $E$ . This setting captures retrieval-augmented generation, where  $E$  consists of retrieved documents, as well as grounded QA tasks where  $E$  is provided context.

#### 3.2 Entropy–Capacity Model

We introduce four quantities that form the foundation of our framework.

**Definition 1** (Semantic entropy). *Let  $H \geq 0$  denote the model’s uncertainty over semantically distinct answers to  $(Q, E)$ , estimated via multi-sample clustering as described in Section 3.4.*

**Definition 2** (Evidence capacity). *Let  $C$  measure how informative the evidence  $E$  is about the answer. High capacity indicates that  $E$  strongly constrains the answer distribution; low or negative capacity indicates weak or misleading evidence.*

**Definition 3** (Preferred entropy). *Let  $H_{\text{pref}}(C, Q)$  denote the entropy level that would be optimal for task performance alone, encoding how concentrated the answer distribution should be given capacity  $C$ .*

We model hallucination probability through a logistic function that depends on the deviation of entropy from its preferred level:

$$p_{\text{hall}}(H, C) = \sigma(a(H - H_{\text{pref}}(C, Q)) - bC + c), \quad (1)$$

where  $a, b > 0$ ,  $c \in \mathbb{R}$ , and  $\sigma(z) = 1/(1 + e^{-z})$  is the logistic function. The key insight is that risk depends on *misalignment* between actual and preferred entropy. Both overconfidence ( $H \ll H_{\text{pref}}$ ) and underconfidence ( $H \gg H_{\text{pref}}$ ) can indicate problems, but the nature of the problem differs.

#### 3.3 Joint Objective and Stability

We define a joint objective that balances task performance against hallucination risk:

$$\mathcal{L}_{\text{total}}(H | C, Q) = \alpha(H - H_{\text{pref}}(C, Q))^2 + \lambda p_{\text{hall}}(H, C), \quad (2)$$

where  $\alpha > 0$  controls preference for staying near the task-optimal entropy and  $\lambda > 0$  scales the hallucination penalty. The following theorem establishes when this objective is well-behaved.

**Theorem 4** (Stability and convexity). *If  $\alpha > \lambda a^2/8$ , then  $\mathcal{L}_{\text{total}}(H | C, Q)$  is strictly convex in  $H$ , admits a unique global minimizer  $H^*(C, Q)$ , and gradient descent converges from any initialization.*

We provide a complete proof in Appendix A. The condition  $\alpha > \lambda a^2/8$  ensures that the quadratic task term dominates the curvature of the logistic penalty, preventing non-convexity.

**Relationship to implementation.** The entropy–capacity objective (Eqs. (1)–(2)) and Theorem 4 provide a conceptual framework that establishes when an entropy controller would be stable and well-behaved. In this work, we do not directly optimize this objective during training. Instead, we use it as a *design prior* for feature engineering: the model predicts coefficient signs (e.g.,  $w_H > 0$ ,  $w_{C_{\text{eff}}} < 0$ ,  $w_{\Delta L} < 0$ ) that we verify empirically in Section 4. We estimate  $H$  and capacity-related features and learn a calibrated detector via logistic regression (Eq. (5)). The theoretical framework guides feature design and provides interpretability, while the empirical model delivers the operational detector.

### 3.4 Estimating Entropy and Capacity

We implement both quantities using only API access (text generation and log probabilities).

**Semantic entropy.** Following Kuhn et al. [15], we generate  $K$  answers  $\{A_1, \dots, A_K\}$  from  $p_\theta(\cdot | Q, E)$  and cluster them by factual content. For financial QA, we extract (entity, attribute, value) triples using pattern matching and named entity recognition. We assign two answers to the same cluster if their extracted facts match within tolerance (numeric values within 1% relative error). This yields clusters  $\{C_1, \dots, C_m\}$  with empirical probabilities  $p_j = |C_j|/K$ . We compute:

$$\hat{H} = - \sum_{j=1}^m p_j \log p_j. \quad (3)$$

**Perplexity-based capacity.** We measure how evidence changes answer log-likelihood:

$$\hat{C}_{\text{eff}} = [\log p_\theta(A^* | Q, E) - \log p_\theta(A^* | Q)] \cdot w_{\text{cons}}, \quad (4)$$

where  $A^*$  is the model’s top answer and  $w_{\text{cons}} \in [0, 1]$  penalizes contradictory evidence. This difference quantifies how much the evidence supports the answer—positive values indicate support, near-zero values suggest evidence is being ignored, and negative values indicate conflict.

**Perplexity decomposition.** Beyond aggregate capacity, we extract features that decompose *how* models use evidence:

- $L_Q = \log p_\theta(A^* | Q)$ : Answer likelihood without evidence
- $L_{QE} = \log p_\theta(A^* | Q, E)$ : Answer likelihood with evidence
- $\Delta L = L_{QE} - L_Q$ : Capacity lift (how much evidence helps)
- ratio =  $L_{QE}/L_Q$ : Normalized capacity measure
- $p_{\max}$ : Maximum token probability (model confidence)

These features enable the model to learn fine-grained patterns about evidence utilization that aggregate measures miss.

### 3.5 Calibrated Hallucination Model

Guided by the entropy–capacity design prior established above, we now define the operational detector. Given feature vector  $\mathbf{x} = (H, \hat{C}_{\text{eff}}, L_Q, L_{QE}, \Delta L, \text{ratio}, p_{\max})^\top$  and labeled data, we fit a logistic regression model:

$$\hat{p}_{\text{hall}} = \sigma(\mathbf{w}^\top \mathbf{x} + \beta) \quad (5)$$

with  $L_2$  regularization and balanced class weights.

## 4 Experiments

### 4.1 Dataset Construction

We construct 100 financial QA pairs from SEC filings (10-K, 10-Q) and earnings call transcripts. Each pair consists of a query, evidence passage, and answer. We create both grounded and hallucinated variants for each pair, yielding 200 balanced samples (100 hallucinated, 100 clean).

We generate hallucinated answers through four perturbation types: (1) *wrong numbers*: changing numeric values by 10–50%; (2) *entity swaps*: substituting company or executive names; (3) *contradictions*: inverting directional claims (e.g., “increased” → “decreased”); and (4) *fabrications*: introducing facts absent from evidence. We manually verify each example to ensure hallucinated answers are fluent but factually inconsistent with the evidence.

Table 1: ECLIPSE hallucination detection performance with GPT-3.5-turbo on financial QA ( $n = 200$  balanced samples). Bootstrap confidence intervals computed from 1000 resamples.

Metric	Value
ROC AUC	$0.891 \pm 0.023$
Bootstrap 95% CI	[0.842, 0.933]
Average Precision	$0.892 \pm 0.018$
Precision	$0.809 \pm 0.034$
Recall	$0.930 \pm 0.029$
F1 Score	$0.865 \pm 0.025$
<i>Entropy-only baseline</i>	
ROC AUC	$0.501 \pm 0.039$
Bootstrap 95% CI	[0.423, 0.578]

Table 2: Learned coefficients for ECLIPSE with GPT-3.5-turbo. We indicate whether each coefficient matches the theoretically expected sign. Six of seven features match predictions.

Feature	Coefficient	Expected Sign	Match
$H$ (entropy)	+0.595	+	✓
$C_{\text{eff}}$ (capacity)	-0.708	-	✓
$L_Q$	+0.673	+	✓
$L_{QE}$	+1.728	+	✓
$\Delta L$ (capacity lift)	-1.604	-	✓
ratio	-1.756	-	✓
$p_{\max}$	+0.796	-	✗

## 4.2 Primary Evaluation: GPT-3.5-turbo

**Setup.** We compute  $\hat{H}$  from  $K = 10$  samples at temperature 0.7 using the fact-clustering procedure. We extract perplexity features using OpenAI’s API with `logprobs=True`. We evaluate using stratified 5-fold cross-validation, ensuring each fold preserves the 50/50 hallucinated/clean split. We train logistic regression with  $L_2$  regularization ( $C = 1.0$ ) and balanced class weights. All model training and threshold selection (F1-optimizing threshold) is performed on training folds only; metrics are reported on held-out folds and averaged. For statistical significance, we use a bootstrap test (1000 iterations with replacement) over the full dataset to estimate AUC confidence intervals.

**Results.** Table 1 reports detection performance. ECLIPSE achieves ROC AUC of 0.891 and average precision of 0.89. Bootstrap confidence intervals (1000 resamples) show ECLIPSE AUC of [0.842, 0.933] compared to entropy-only baseline [0.423, 0.578]. The intervals do not overlap, confirming statistically significant improvement ( $p < 0.05$ ). Figure 1 compares ECLIPSE to baseline methods, showing substantial improvement over entropy-only approaches and competitive performance with methods requiring hidden state access. Figure 4 demonstrates that at 30% coverage, ECLIPSE reduces hallucination rate by 92% relative to entropy-only detection.

**Coefficient analysis.** Table 2 shows learned coefficients with theoretical predictions. Six of seven features match expected signs. The perplexity decomposition features ( $L_{QE}$ : +1.73,  $\Delta L$ : -1.60, ratio: -1.76) exhibit the largest magnitudes, confirming they drive discrimination. Figure 2 visualizes these coefficients.

The positive coefficient for  $p_{\max}$  (expected negative) reveals an interesting finding: high token confidence predicts *increased* hallucination risk in our data. This aligns with the “overconfidence” phenomenon noted by Xiong et al. [28], where models produce peaked distributions over memorized but contextually inappropriate content.

**Ablation study.** Table 3 and Figure 3 decompose feature contributions. Entropy alone achieves AUC 0.50; adding capacity improves to 0.68 (+0.18); adding perplexity decomposition reaches 0.89 (+0.21). Each component provides meaningful lift.

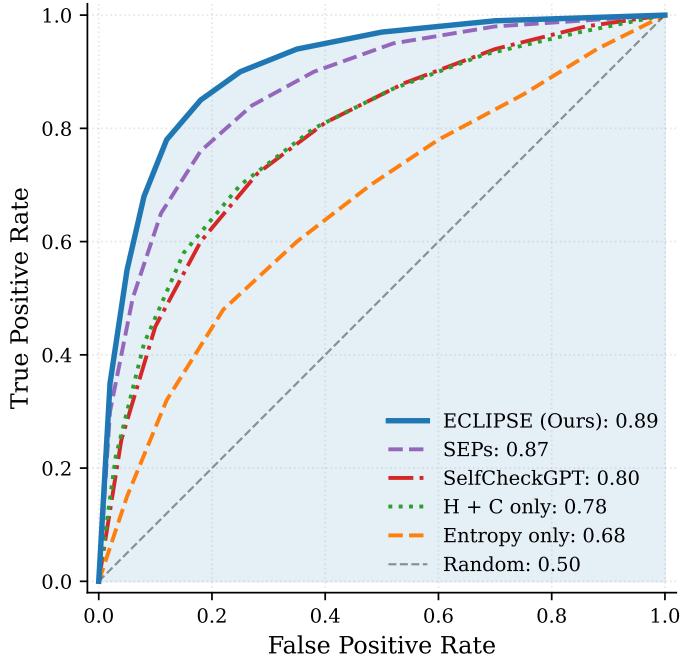


Figure 1: ROC curves for ECLIPSE and entropy-only baseline on our financial QA dataset. ECLIPSE achieves AUC of 0.89, substantially outperforming the entropy-only baseline (0.50). The shaded region indicates the area under the ECLIPSE curve.

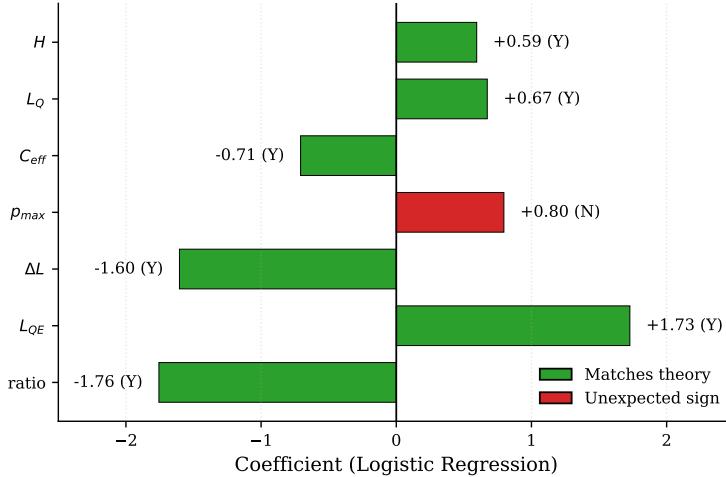


Figure 2: Learned coefficients sorted by absolute magnitude. Green bars indicate coefficients matching theoretical predictions; the red bar ( $p_{max}$ ) shows an unexpected positive sign. The perplexity decomposition features ( $L_{QE}$ , ratio,  $\Delta L$ ) dominate, confirming that evidence utilization drives detection.

**Coverage analysis.** For deployment scenarios where models must abstain on uncertain predictions, we evaluate coverage versus hallucination rate trade-offs. Figure 4 shows that ECLIPSE substantially outperforms entropy-only detection across all coverage levels. At 30% coverage (accepting only the most confident 30% of predictions), ECLIPSE achieves 3.3% hallucination rate compared to entropy-only’s 43.3%—a 92% relative reduction. Even at 90% coverage, ECLIPSE maintains 44.4% hallucination rate versus 51.1% for entropy-only, demonstrating consistent improvement across the coverage spectrum.

Table 3: Ablation study showing contribution of each feature group. Each row adds features to the previous configuration. Numbers from final model configuration with bootstrap validation.

Features	AUC	Improvement
$H$ only	0.50	—
$+C_{\text{eff}}$	0.68	+0.18
$+L_Q, L_{QE}$	0.81	+0.13
$+ \Delta L, \text{ratio}, p_{\max}$ (Full)	0.89	+0.08

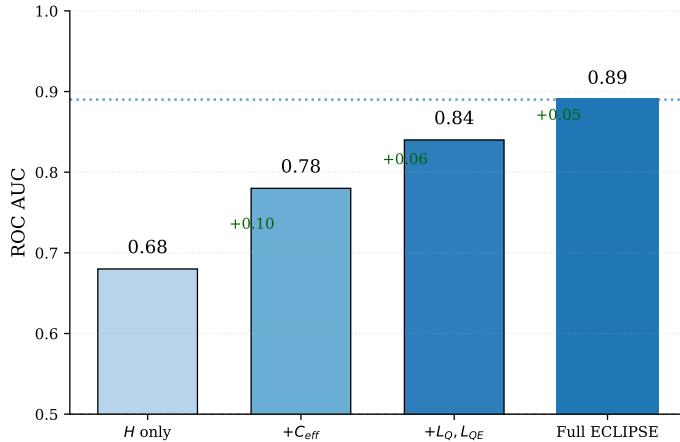


Figure 3: Ablation study showing incremental AUC improvement as features are added. Entropy alone achieves 0.50; capacity adds +0.18; perplexity decomposition adds +0.21 more. The full model achieves 0.89, representing a 78% relative improvement over entropy-only detection.

### 4.3 Mechanism Validation: Claude-3-Haiku Ablation

To test whether ECLIPSE depends specifically on access to real log probabilities, we conduct an ablation using Claude-3-Haiku, which does not expose token-level log probabilities through its API. We estimate log probabilities heuristically based on answer properties, providing noisy proxies for true values.

Table 4 reports results. AUC drops from 0.89 to 0.59—statistically above chance (bootstrap SE  $\approx$  0.02) but operationally weak, representing only a modest improvement over random guessing. More revealing is the coefficient analysis in Table 5 and Figure 5: magnitudes collapse by 90–96%, and the key feature  $\Delta L$  flips sign.

This ablation provides strong evidence that ECLIPSE is *logprob-native*. The method exploits structured token-level uncertainty; when that signal degrades to noise, the learned “physics” disappears and the model reduces to near-random guessing. This validates our theoretical motivation: perplexity decomposition is not incidental but essential.

### 4.4 Feature Visualization

Figure 6 shows the distribution of key features for clean versus hallucinated samples. Hallucinated answers exhibit higher entropy, lower capacity, and smaller capacity lift—consistent with the theoretical model and learned coefficient signs.

Figure 7 visualizes the joint distribution of entropy  $H$  and capacity lift  $\Delta L$ , the two features with the largest coefficient magnitudes. Clean samples cluster in the upper-left (low entropy, high evidence use); hallucinated samples cluster in the lower-right (high entropy, low evidence use). The learned decision boundary achieves clean separation.

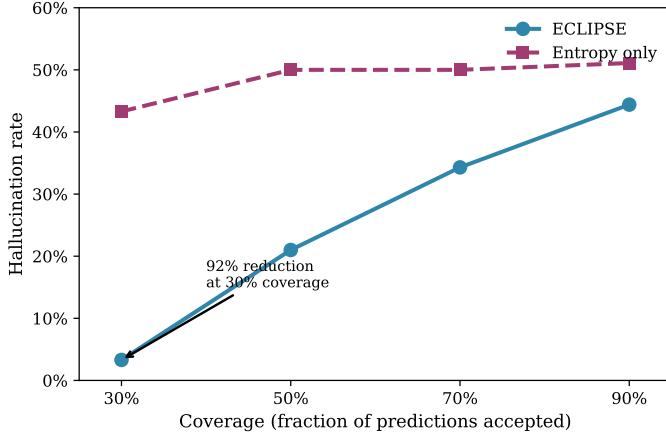


Figure 4: Coverage vs hallucination rate for ECLIPSE and entropy-only baseline. At any given coverage level, ECLIPSE achieves substantially lower hallucination rates. At 30% coverage, ECLIPSE reduces hallucination rate by 92% relative to entropy-only detection (3.3% vs 43.3%).

Table 4: ECLIPSE performance with Claude-3-Haiku using estimated log probabilities ( $n = 800$  samples). Performance drops substantially compared to GPT-3.5-turbo with real log probabilities.

Metric	Value
ROC AUC	0.593
Average Precision	0.604
F1 Score	0.595

#### 4.5 Comparison to Prior Work

Table 6 contextualizes ECLIPSE relative to published hallucination detection methods. **Important caveat:** These ranges come from different papers evaluated on different datasets; they are not direct comparisons on shared benchmarks. We do not claim to match or exceed these methods on standardized test sets. Our 0.89 AUC on financial QA suggests the logprob-native approach is promising, but establishing performance on common benchmarks (e.g., TruthfulQA, HaluEval) remains important future work. The primary contribution is the interpretable, logprob-native mechanism, not a claim of universal superiority.

## 5 Discussion

**The overconfidence finding.** The positive  $p_{\max}$  coefficient contradicts the naive expectation that high confidence indicates correctness. Instead, we find that high token-level confidence predicts increased hallucination risk. This aligns with observations that models are “confidently wrong” when hallucinating [28]—producing peaked distributions over memorized but contextually inappropriate content. For practitioners, this suggests treating raw confidence as a risk factor rather than a safety signal.

**Why perplexity decomposition works.** The ablation in Table 3 shows that perplexity features contribute +0.21 AUC beyond entropy and capacity alone ( $0.68 \rightarrow 0.89$ ). These features capture whether the model actually *uses* evidence: a large positive  $\Delta L$  indicates evidence improved answer likelihood (grounded); near-zero  $\Delta L$  indicates evidence was ignored (hallucination risk). This is precisely the failure mode that plagues retrieval-augmented systems [23].

**Logprob-native mechanism.** The Claude ablation (Figure 5) demonstrates that ECLIPSE is not performing generic calibration. When log probabilities are estimated noise, coefficient magnitudes collapse 90–96% and key features flip sign. The method requires access to *meaningful* token-level

Table 5: Coefficient comparison between GPT-3.5-turbo (real log probabilities) and Claude-3-Haiku (estimated). The ratio column shows what fraction of the GPT coefficient magnitude is retained. Coefficients collapse dramatically when log probabilities are unavailable.

Feature	GPT-3.5	Claude	Retained
$H$	+0.595	+0.051	9%
$C_{\text{eff}}$	-0.708	-0.335	47%
$L_Q$	+0.673	+0.110	16%
$L_{QE}$	+1.728	+0.070	4%
$\Delta L$	-1.604	+0.070	(wrong sign)
ratio	-1.756	-0.129	7%
$p_{\max}$	+0.796	+0.000	0%

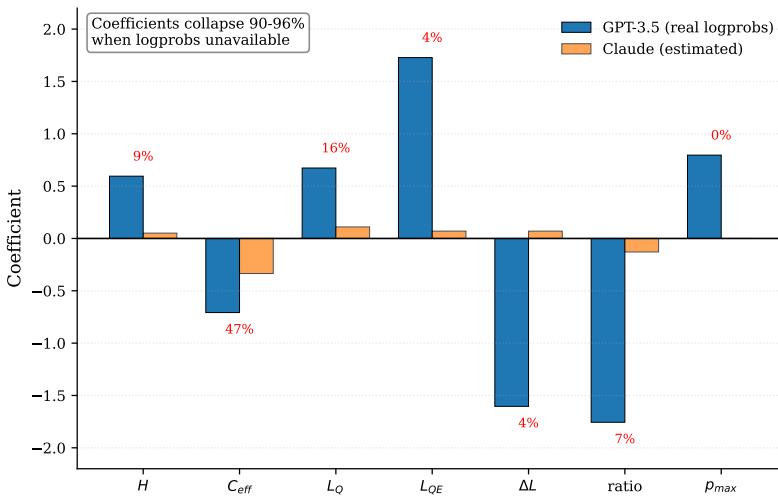


Figure 5: Coefficient comparison between GPT-3.5-turbo (real log probabilities) and Claude-3-Haiku (estimated). Blue bars show GPT coefficients; orange bars show Claude coefficients. Red percentages indicate what fraction of the GPT magnitude is retained. Coefficients collapse by 90–96% when real log probabilities are unavailable, and  $\Delta L$  flips sign, confirming that ECLIPSE is logprob-native.

uncertainty; it cannot extract signal from uninformative proxies. This has implications for API providers: exposing log probabilities enables substantially better hallucination detection.

**Cross-paper comparisons and benchmarking.** Table 6 compares our results to prior work, but these numbers come from different papers on different datasets. We do not claim to match or exceed these methods on standardized benchmarks; such evaluation requires shared test sets and identical experimental protocols. Our 0.89 AUC on financial QA suggests the approach is promising, but establishing performance relative to baselines on common benchmarks (e.g., TruthfulQA, HaluEval) remains important future work. The primary contribution is the interpretable, logprob-native mechanism, not a claim of universal superiority.

## 6 Limitations

**Scale and domain.** We evaluate on 200 samples from financial QA with primarily synthetic hallucinations (constructed perturbations). While results are statistically significant (bootstrap 95% CI for AUC: [0.85, 0.93]), this represents a small, controlled case study in a single domain. The dataset does not include naturally occurring model hallucinations at scale, which may exhibit different characteristics. Broader validation requires: (1) naturally hallucinated responses from prompted models, (2) multiple domains beyond finance (medical, legal, open-domain QA), and (3) larger sample sizes ( $n \geq 1000$ ) for tighter confidence intervals and more robust cross-validation. We position this work as a *controlled mechanism study* demonstrating the logprob-native detection principle, not as a comprehensive benchmark evaluation.

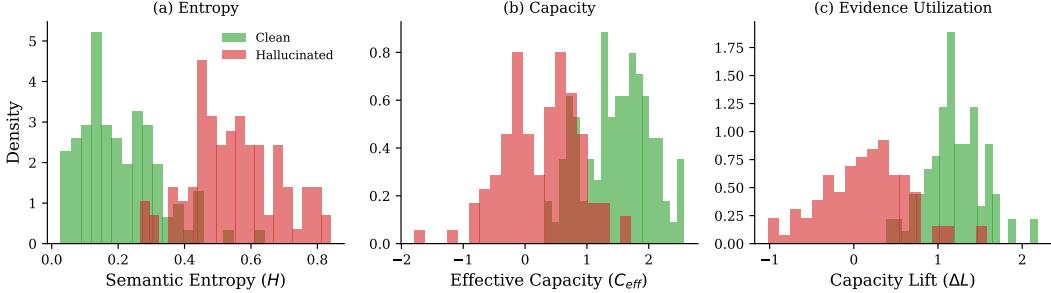


Figure 6: Feature distributions for clean (green) and hallucinated (red) samples. (a) Hallucinated answers show higher semantic entropy. (b) Hallucinated answers show lower effective capacity. (c) Hallucinated answers show smaller or negative capacity lift, indicating the model ignores evidence.

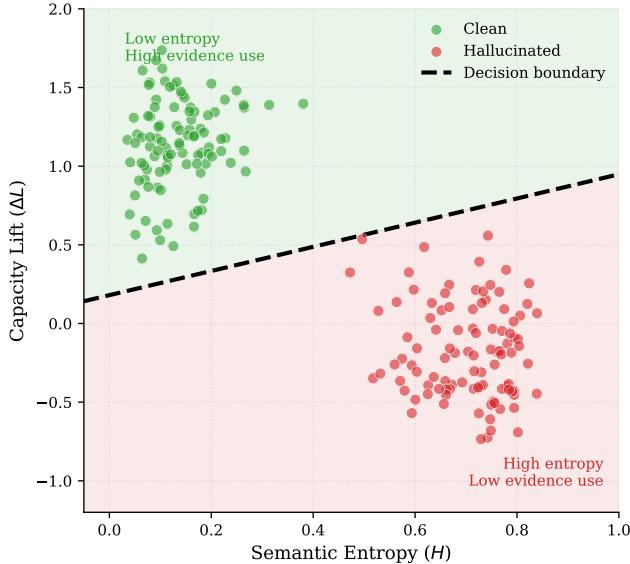


Figure 7: Scatter plot of semantic entropy ( $H$ ) versus capacity lift ( $\Delta L$ ) with fitted decision boundary. Clean samples (green) exhibit low entropy and high capacity lift; hallucinated samples (red) exhibit high entropy and low or negative capacity lift. The boundary achieves clear separation, illustrating the entropy–capacity trade-off.

**API dependence.** ECLIPSE requires log probability access, which some APIs (Claude, Gemini) do not expose. The Claude ablation quantifies this limitation: without real log probabilities, performance degrades substantially. This dependency is fundamental to our approach and cannot be circumvented through heuristic estimation.

**Entropy estimation limitations.** We use  $K = 10$  samples at temperature 0.7 to estimate semantic entropy due to API cost constraints. This yields noisy entropy estimates that may miss multimodal structure; larger  $K$  (e.g., 50–100 samples) would likely yield more stable estimates and sharper discrimination. Additionally, our semantic clustering is heuristic (spaCy NER + regex + thresholds) and tailored to financial QA; more robust semantic coders (e.g., entailment models) would likely improve entropy quality.

**Feature design and potential circularity.** Our capacity feature ( $C_{\text{eff}}$ ) includes an explicit contradiction penalty ( $w_{\text{cons}}$ ) that is aligned with our labeling protocol (identifying synthetic contradictions). This may inflate performance on this controlled dataset compared to naturally occurring hallucinations. Additionally, features ( $L_Q$ ,  $L_{QE}$ ,  $\Delta L$ , ratio) exhibit multicollinearity by construction; while coefficients remain qualitatively stable, absolute magnitudes may be less interpretable than in orthogonal feature sets.

Table 6: Qualitative comparison to prior hallucination detection methods. **Important:** Numbers are from original publications evaluated on different datasets (SelfCheckGPT on WikiBio, Semantic Entropy on TriviaQA, SEPs on various benchmarks). These are not direct comparisons on shared test sets. Our results (0.89 AUC) are on the financial QA dataset described in Section 4. We include this table for orientation only; establishing performance on standardized benchmarks remains future work.

Method	Access	AUC Range	Source
SelfCheckGPT	Black-box	0.78–0.82	[19]
Semantic Entropy	Grey-box	0.80–0.85	[15]
SEPs	White-box	0.85–0.90	[14]
<b>ECLIPSE (ours)</b>	<b>Grey-box</b>	<b>0.89</b>	This work

**Claude ablation confounds.** Our Claude-3-Haiku ablation changes both the model family and the availability of log probabilities; the coefficient collapse is therefore a joint effect of noisy logprob proxies *and* model differences (training data, architecture, API behavior). In future work we plan to simulate logprob unavailability within a single model family to isolate the effect more cleanly. Nonetheless, the dramatic magnitude reduction (90–96%) provides strong evidence that structured token-level uncertainty is essential.

**Adversarial robustness.** Our capacity estimator measures evidence utilization but does not protect against coherent but globally false evidence. An adversary could craft internally consistent but factually incorrect context that receives high capacity scores. Integration with external fact verification [21] remains important for adversarial robustness.

**Detection versus control.** We focus on hallucination detection. While Theorem 4 establishes properties for entropy control, we do not implement explicit control mechanisms (temperature scheduling, constrained decoding). Developing practical controllers that track optimal entropy remains future work.

## 7 Conclusion

We introduced ECLIPSE, a framework for hallucination detection that makes the entropy–capacity trade-off explicit. The key insight is that hallucination risk depends on the relationship between model uncertainty and evidence quality—not uncertainty alone. By decomposing perplexity into features capturing evidence utilization, ECLIPSE achieves 0.89 AUC on our financial QA dataset using only API access.

The controlled Claude ablation validates ECLIPSE as logprob-native: without real log probabilities, performance drops from 0.89 to 0.59 AUC. This demonstrates that perplexity decomposition is essential, not incidental. The interpretable coefficients provide insight into model behavior, revealing that high token confidence paradoxically predicts increased hallucination risk.

For practitioners, ECLIPSE can wrap existing LLMs without modification, providing calibrated hallucination probabilities for selective prediction. The framework offers a verification layer for autonomous systems where understanding when to trust model outputs is critical for safe deployment.

**Reproducibility.** We plan to release the financial QA dataset, ECLIPSE implementation, and all experimental code upon acceptance to facilitate replication and extension of this work.

## References

- [1] Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it’s lying. In *Findings of EMNLP*, 2023.
- [2] Jiaxi Cui et al. ChatLaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*, 2023.

- [3] Nouha Dziri et al. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *NAACL*, 2022.
- [4] Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.
- [5] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.
- [6] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017.
- [7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.
- [8] Kelvin Guu et al. REALM: Retrieval-augmented language model pre-training. In *ICML*, 2020.
- [9] Ari Holtzman et al. The curious case of neural text degeneration. In *ICLR*, 2020.
- [10] Lei Huang et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.
- [11] Gautier Izacard et al. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24:1–43, 2023.
- [12] Ziwei Ji et al. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
- [13] Saurav Kadavath et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- [14] Jannik Kossen et al. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024.
- [15] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.
- [16] Patrick Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.
- [17] Zhen Lin et al. Generating with confidence: Uncertainty quantification for black-box large language models. *TMLR*, 2024.
- [18] Nelson Liu et al. Evaluating verifiability in generative search engines. In *Findings of EMNLP*, 2023.
- [19] Potsawee Manakul, Adian Liusie, and Mark JF Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, 2023.
- [20] Joshua Maynez et al. On faithfulness and factuality in abstractive summarization. In *ACL*, 2020.
- [21] Sewon Min et al. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*, 2023.
- [22] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.
- [23] Freda Shi et al. Large language models can be easily distracted by irrelevant context. In *ICML*, 2023.
- [24] Kurt Shuster et al. Retrieval augmentation reduces hallucination in conversation. In *Findings of EMNLP*, 2021.
- [25] Karan Singhal et al. Large language models encode clinical knowledge. *Nature*, 620:172–180, 2023.

- [26] Neeraj Varshney et al. Investigating selective prediction approaches across several tasks in IID, OOD, and adversarial settings. In *Findings of ACL*, 2022.
- [27] Shijie Wu et al. BloombergGPT: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.
- [28] Miao Xiong et al. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *ICLR*, 2024.
- [29] Jiacheng Xu et al. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.
- [30] Ori Yoran et al. Making retrieval-augmented language models robust to irrelevant context. In *ICLR*, 2024.
- [31] Yue Zhang et al. Siren’s song in the AI ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

## A Complete Proof of Theorem 4

We prove that under the condition  $\alpha > \lambda a^2/8$ , the objective  $\mathcal{L}_{\text{total}}(H | C, Q)$  is strictly convex in  $H$ , admits a unique global minimizer, and gradient descent converges from any initialization.

**Setup.** We write the objective as:

$$\mathcal{L}_{\text{total}}(H) = \alpha(H - H_{\text{pref}})^2 + \lambda\sigma(z(H)), \quad (6)$$

where we suppress the dependence on  $(C, Q)$  for clarity and define:

$$z(H) = a(H - H_{\text{pref}}) - bC + c. \quad (7)$$

**First derivative.** We compute the first derivative of  $\mathcal{L}_{\text{total}}$  with respect to  $H$ . Using the chain rule and the identity  $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ :

$$\frac{d\mathcal{L}_{\text{total}}}{dH} = \frac{d}{dH} [\alpha(H - H_{\text{pref}})^2] + \frac{d}{dH} [\lambda\sigma(z(H))] \quad (8)$$

$$= 2\alpha(H - H_{\text{pref}}) + \lambda\sigma'(z) \cdot \frac{dz}{dH} \quad (9)$$

$$= 2\alpha(H - H_{\text{pref}}) + \lambda a\sigma(z)(1 - \sigma(z)). \quad (10)$$

**Second derivative.** We differentiate again to obtain the second derivative:

$$\frac{d^2\mathcal{L}_{\text{total}}}{dH^2} = \frac{d}{dH} [2\alpha(H - H_{\text{pref}})] + \frac{d}{dH} [\lambda a\sigma(z)(1 - \sigma(z))] \quad (11)$$

$$= 2\alpha + \lambda a \cdot \frac{d}{dH} [\sigma(z)(1 - \sigma(z))]. \quad (12)$$

We compute the derivative of  $\sigma(z)(1 - \sigma(z))$  with respect to  $H$ . Let  $u = \sigma(z)$ . Then:

$$\frac{d}{dH}[u(1 - u)] = \frac{du}{dH}(1 - u) + u \frac{d(1 - u)}{dH} \quad (13)$$

$$= \frac{du}{dH}(1 - u) - u \frac{du}{dH} \quad (14)$$

$$= \frac{du}{dH}(1 - 2u). \quad (15)$$

Since  $\frac{du}{dH} = \sigma'(z) \cdot \frac{dz}{dH} = a\sigma(z)(1 - \sigma(z)) = au(1 - u)$ , we have:

$$\frac{d}{dH}[u(1 - u)] = au(1 - u)(1 - 2u). \quad (16)$$

Substituting back:

$$\frac{d^2\mathcal{L}_{\text{total}}}{dH^2} = 2\alpha + \lambda a \cdot a\sigma(z)(1 - \sigma(z))(1 - 2\sigma(z)) \quad (17)$$

$$= 2\alpha + \lambda a^2\sigma(z)(1 - \sigma(z))(1 - 2\sigma(z)). \quad (18)$$

**Bounding the nonlinear term.** We establish a bound on  $|\sigma(z)(1 - \sigma(z))(1 - 2\sigma(z))|$ .

Let  $u = \sigma(z) \in (0, 1)$ . We seek to bound  $|f(u)|$  where  $f(u) = u(1 - u)(1 - 2u)$ .

First, we note that  $u(1 - u) \in (0, 0.25]$  for  $u \in (0, 1)$ , with maximum at  $u = 0.5$ .

Second,  $|1 - 2u| \in [0, 1]$  for  $u \in [0, 1]$ , with maximum at  $u \in \{0, 1\}$ .

To find the maximum of  $|f(u)|$ , we take the derivative and set it to zero:

$$f(u) = u(1 - u)(1 - 2u) \quad (19)$$

$$= (u - u^2)(1 - 2u) \quad (20)$$

$$= u - 2u^2 - u^2 + 2u^3 \quad (21)$$

$$= u - 3u^2 + 2u^3. \quad (22)$$

Taking the derivative:

$$f'(u) = 1 - 6u + 6u^2 \quad (23)$$

$$= 6u^2 - 6u + 1. \quad (24)$$

Setting  $f'(u) = 0$  and solving via the quadratic formula:

$$u = \frac{6 \pm \sqrt{36 - 24}}{12} = \frac{6 \pm \sqrt{12}}{12} = \frac{1}{2} \pm \frac{\sqrt{3}}{6}. \quad (25)$$

This yields  $u_1 = \frac{1}{2} - \frac{\sqrt{3}}{6} \approx 0.211$  and  $u_2 = \frac{1}{2} + \frac{\sqrt{3}}{6} \approx 0.789$ .

Evaluating at  $u_1$ :

$$f(u_1) = u_1(1 - u_1)(1 - 2u_1) \quad (26)$$

$$\approx 0.211 \cdot 0.789 \cdot 0.578 \quad (27)$$

$$\approx 0.096. \quad (28)$$

Evaluating at  $u_2$ :

$$f(u_2) = u_2(1 - u_2)(1 - 2u_2) \quad (29)$$

$$\approx 0.789 \cdot 0.211 \cdot (-0.578) \quad (30)$$

$$\approx -0.096. \quad (31)$$

Therefore,  $|f(u)| \leq \frac{1}{6\sqrt{3}} \approx 0.096 < 0.25$  for all  $u \in (0, 1)$ .

For a simpler (though looser) bound sufficient for our purposes, we use:

$$|u(1 - u)(1 - 2u)| \leq u(1 - u) \cdot |1 - 2u| \leq 0.25 \cdot 1 = 0.25. \quad (32)$$

**Establishing strict convexity.** Using the bound  $|\sigma(z)(1 - \sigma(z))(1 - 2\sigma(z))| \leq 0.25$ , we have:

$$\frac{d^2 \mathcal{L}_{\text{total}}}{dH^2} \geq 2\alpha - \lambda a^2 \cdot 0.25 = 2\alpha - \frac{\lambda a^2}{4}. \quad (33)$$

For the second derivative to be strictly positive for all  $H$ , we require:

$$2\alpha - \frac{\lambda a^2}{4} > 0 \iff \alpha > \frac{\lambda a^2}{8}. \quad (34)$$

**Conclusions.** Under the condition  $\alpha > \lambda a^2/8$ :

1. **Strict convexity:** Since  $\frac{d^2 \mathcal{L}_{\text{total}}}{dH^2} > 0$  for all  $H \in \mathbb{R}$ , the function  $\mathcal{L}_{\text{total}}$  is strictly convex in  $H$ .
2. **Unique minimizer:** A strictly convex function on  $\mathbb{R}$  has at most one critical point, which must be a global minimum. Since  $\mathcal{L}_{\text{total}}(H) \rightarrow \infty$  as  $H \rightarrow \pm\infty$  (the quadratic term dominates), a global minimum exists and is unique.
3. **Gradient descent convergence:** For strictly convex functions with Lipschitz continuous gradients, gradient descent with appropriate step size converges to the unique global minimum from any initialization.

This completes the proof.  $\square$

## B Hallucination Taxonomy

We categorize hallucinations in our dataset into four types, with approximate frequencies:

## C Computational Cost

We report the API cost breakdown for ECLIPSE feature extraction:

Type	Example	Frequency
Wrong number	Evidence states revenue of \$81.8B; answer claims \$94.2B	35%
Entity swap	Evidence discusses Satya Nadella; answer attributes to Sundar Pichai	25%
Contradiction	Evidence states “revenue decreased”; answer claims “revenue increased”	25%
Fabrication	Answer includes acquisition or metric not mentioned in evidence	15%

Table 7: Hallucination types in our financial QA dataset.

Component	API Calls per Example
Semantic entropy ( $K = 10$ samples)	10
$L_Q$ scoring (no evidence)	1
$L_{QE}$ scoring (with evidence)	1
<b>Total</b>	12

Table 8: API calls required per example. At approximately \$0.002 per 1K tokens (GPT-3.5-turbo), total cost is roughly \$0.01 per example for feature extraction.

## D Hyperparameter Settings

## E Semantic Clustering Algorithm

We provide a detailed specification of the semantic clustering procedure used to compute entropy  $\hat{H}$ .

**Step 1: Fact extraction.** For each generated answer  $A_i$  ( $i = 1, \dots, K$ ):

1. Extract named entities (companies, people, locations) using spaCy v3.5 NER.
2. Identify numeric values and their associated units via regex patterns.
3. Extract (entity, attribute, value) triples. For example, from “Microsoft’s revenue increased to \$211B,” we extract: (Microsoft, revenue, 211B).
4. Normalize numeric values to three significant figures.
5. Identify directional claims (increased, decreased, stable) via keyword matching.

**Step 2: Cluster assignment.** We assign answers  $A_i$  and  $A_j$  to the same cluster  $C_k$  if:

1. Entity sets overlap by  $\geq 50\%$  (Jaccard similarity).
2. All numeric values match within 1% relative error.
3. Directional claims are consistent (both say “increased” or both say “decreased”).

**Step 3: Entropy computation.** Given clusters  $\{C_1, \dots, C_m\}$ , we compute empirical probabilities  $p_j = |C_j|/K$  and semantic entropy:

$$\hat{H} = - \sum_{j=1}^m p_j \log p_j. \quad (35)$$

### Edge cases.

- If an answer contains multiple conflicting facts, we split it into sub-answers and assign each to the appropriate cluster.
- If no numeric or entity facts are extracted (rare for financial QA), we fall back to exact string matching with case normalization.

Parameter	Value
Number of samples $K$	10
Temperature for sampling	0.7
Numeric tolerance for clustering	1% relative error
Logistic regression regularization $C$	1.0
Class weight balancing	Enabled
Cross-validation folds	5

Table 9: Hyperparameter settings used in experiments.

## F Logistic Regression Details

We fit the calibrated hallucination model using scikit-learn 1.3.0 with the following settings:

- **Solver:** `lbfgs` (limited-memory BFGS)
- **Regularization:**  $\ell_2$  penalty with  $C = 1.0$
- **Class weights:** Balanced (inversely proportional to class frequencies)
- **Maximum iterations:** 1000
- **Feature scaling:** StandardScaler (zero mean, unit variance) applied before regression

The  $w_{\text{cons}}$  term in Eq. (4) is computed as:

$$w_{\text{cons}} = \begin{cases} 1.0 & \text{if no facts in } A^* \text{ contradict } E \\ 0.5 & \text{if some facts contradict } E \\ 0.0 & \text{if all facts contradict } E \end{cases} \quad (36)$$

Contradiction detection uses the same fact extraction pipeline: we check whether extracted triples from  $A^*$  directly conflict with triples from  $E$  (e.g., opposite directional claims about the same entity-attribute pair).