Detect Before You Disambiguate: Perturbation-Based Input Ambiguity Detection in LLMs

Anonymous ACL submission

Abstract

Large language models (LLMs) often answer ambiguous inputs as if they had a single intended meaning. This is problematic because input ambiguity is an upstream source of uncertainty: when a prompt admits multiple valid interpretations, the key question is whether the model should answer directly at all. Existing input uncertainty methods often rely on LLM-generated statements or clarifications, but such transformations may already disambiguate the input before the ambiguity is measured. We propose a label-preserving perturbation framework for input ambiguity detection in which perturbations remain close to the original input while preserving its label. We instantiate this principle with two detectors: a black-box rule-based perturbation library and a white-box importance-guided method using model-internal signals. Experiments across four datasets in open-domain, conversational, and visual QA show that our methods outperform output-side uncertainty baselines and LLM-paraphrase-based perturbation baselines. Our results highlight the importance of detecting ambiguity before attempting to disambiguate it. Our code is available here.

1 Introduction

Large language models (LLMs) are increasingly deployed as question-answering systems, yet many real-world questions admit more than one legitimate interpretation (Min et al., 2020; Liu et al., 2023; Keyvan and Huang, 2022; Liu et al., 2026). When a question is ambiguous, a model cannot answer it well without first recognizing that an interpretive choice is required. After the ambiguity is recognized, a capable system has several reasonable options: it can ask the user a clarifying question, enumerate the plausible interpretations and answer each, or state the interpretation it is assuming (Zhang et al., 2025; Kuhn et al., 2022). The problematic case is when the model silently commits to a single arbitrary interpretation and generates a single “confident" answer, as if the question had no ambiguity at all. Consider a user who asks "*When did the team win the championship?*" without specifying which team or which championship: a model that quietly settles on one reading may return a fluent, confident year that happens to answer a question the user never asked, with no recognition that other readings exist. Recent studies find that current models frequently fail at exactly this recognition step (Vijayvargiya et al., 2025; Zhang et al., 2024; Kim et al., 2024).

Detecting whether a question is ambiguous is therefore a prerequisite for trustworthy question answering. If ambiguity can be detected, a system need not gamble on an arbitrary interpretation. We therefore frame the problem as a two-stage objective: first, *detect* whether a question is ambiguous, and then, conditioned on that detection, *mitigate* the ambiguity through targeted clarification or enumeration. Reliable detection is the foundation on which mitigation rests.

A natural and widely used strategy for detection is to probe the model’s consistency: rephrase the question several ways, sample answers under each rephrasing, and treat disagreement among the answers as evidence of ambiguity (Hou et al., 2024; Cole et al., 2023; Kuhn et al., 2023). Intuitively, an unambiguous question should yield consistent answers while an ambiguous one should not. However, the existing implementation contains a subtle flaw: *When a language model is used to rephrase the question, it must first understand the question, and understanding an ambiguous question entails resolving its ambiguity*. The rephrased question tends to encode a single interpretation chosen by the rephrasing model, eliminating the ambiguity that the procedure is meant to detect.

Our core principle is that the step of ambiguity detection should not involve understanding the question. To preserve the ambiguity signal,

the transformation applied to the input must be semantics-blind: it should stay close enough to the original input that it does not become a disambiguation step, yet remains expressive enough to expose interpretation-dependent behavior. We model this trade-off in the language of the information bottleneck (Tishby et al., 1999): we seek a perturbation that discards information about the input's surface form while preserving information about its ambiguity, and we use this principle to motivate our method rather than to strictly optimize it.

We instantiate this principle with two complementary detectors. (1) The white-box detector, which assumes access to model internals, uses internal signals to target a set of important input tokens before perturbing them. (2) The black-box detector, which assumes no access to model internals, applies a library of perturbation rules and measures the dispersion in the model's answer distribution. Finally, we use the detection output to drive a lightweight re-asking mitigation strategy.

We evaluate on various datasets spanning open-domain, conversational, and visual question answering. We show that perturbation-based detection more reliably separates ambiguous from unambiguous questions than both output-side sampling and paraphrase-based baselines, with downstream gains when its output is used for mitigation.

2 Related Work

Ambiguity in LLMs Ambiguity is a natural property of user inputs because the same question may license several interpretations. Prior work has studied this issue in open-domain and long-form QA through AmbigQA and ASQA (Min et al., 2020; Stelmakh et al., 2022), and in interactive settings where ambiguity calls for clarification. Abg-CoQA annotates ambiguous conversational questions with clarification turns (Guo et al., 2021), while Abg-SciQA extends ambiguity analysis to scientific questions with labels for ambiguity detection, ambiguity type classification, clarification generation, and clarification-based QA (Chen et al., 2025b). Other benchmarks cover proactive clarification in domain-specific QA (Deng et al., 2022) and multimodal ambiguity in visual QA (Jian et al., 2025; Chen et al., 2025a; Jang and Kim, 2026). Recent LLM-focused analyses further show that ambiguity is not solved simply by scale or prompting (Chen et al., 2025b). LLMs can struggle to solve ambiguous problems (Liu et al., 2023; Zhang et al., 2024; Kamath et al., 2024). To solve the problem, prior work often studies what to do after ambiguity is recognized, including selectively abstaining from ambiguous questions (Cole et al., 2023), prompting models to detect, clarify, and answer (Kuhn et al., 2022), or building clarification-based QA pipelines such as CAmbigNQ (Lee et al., 2023). Our work focuses on detecting whether the original input is ambiguous before resolution. This matters because resolution-oriented transformations, including generated paraphrases, answer sets, or clarification candidates, can potentially choose one interpretation from the ambiguity and remove the ambiguity being measured.

Uncertainty Quantification in LLMs Uncertainty Quantification (UQ) is considered important in trustworthy deployment of LLMs (Liu et al., 2025). Existing UQ methods often estimate reliability from token probabilities, verbalized (Kadavath et al., 2022; Lin et al., 2022; Ren et al., 2023; Azaria and Mitchell, 2023; Chen et al., 2024), but the dominant black-box paradigm for free-form generation is multiple sampling, where several generations are drawn and compared (Chen et al., 2026; Duan et al., 2024; Lin et al., 2024a; Nikitin et al., 2024; Kuhn et al., 2023; Farquhar et al., 2024). For example, Lin et al. (2024b) treats uncertainty as dispersion over possible predictions for a fixed input, and proposes semantic-dispersion measures based on pairwise relations among sampled outputs. Such output-side methods are natural baselines, but diverse generations can reflect either model unreliability or multiple valid interpretations of an ambiguous input. Input-side UQ (Liu et al., 2025) is therefore closer to our setting, and existing methods probe it mainly through prompt perturbations (Hou et al., 2024; Gao et al., 2024; Yang et al., 2024). However, these transformations may remove ambiguity before it is measured. Our framework instead requires label-preserving perturbations, instantiated by a black-box or white-box method, so that measured variance better reflects input ambiguity rather than disambiguation introduced by the perturbation process.

3 Methodology

3.1 Overall Framework

Theoretical motivation. Pure output-side sampling can reveal disagreement among generated answers, but it cannot tell whether the disagreement comes from model unreliability or from multiple

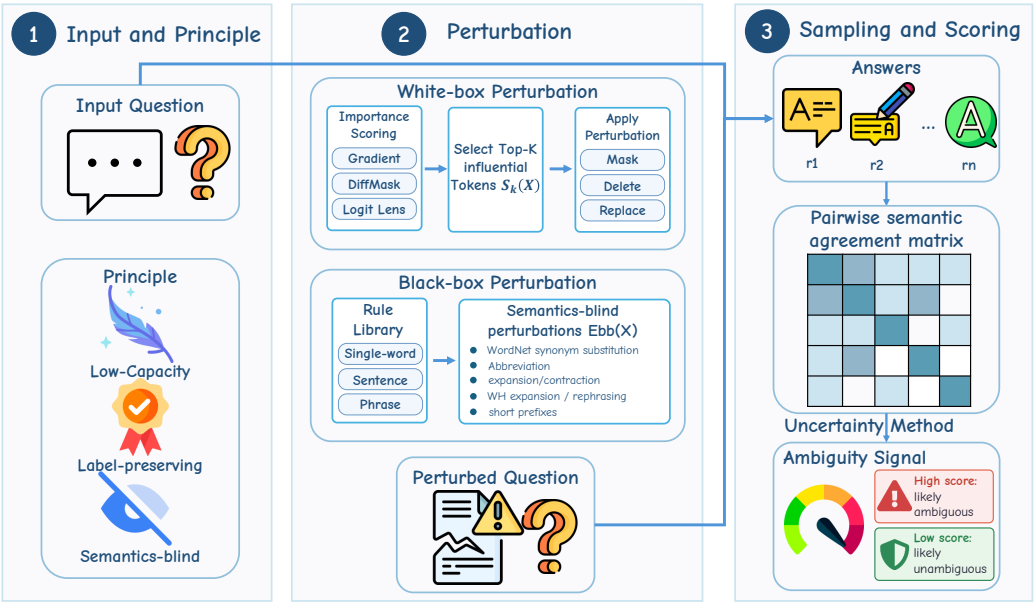


Figure 1: Overview of the proposed ambiguity detection framework. Given an input question and perturbation principles, the method generates label-preserving, low-capacity, and semantics-blind perturbations via white-box token-importance methods and black-box rule-based transformations. The perturbed questions are then sampled to obtain multiple model answers, whose pairwise semantic agreement is used to compute an ambiguity signal: low agreement indicates likely ambiguity, while high agreement suggests the question is likely unambiguous.

valid readings of the input. Direct paraphrasing or clarification generation probes the input side, but the transformation is high-capacity and can resolve the ambiguity before it is measured. To address both issues, our detector asks whether small, label-preserving interventions can expose answer instability without becoming a disambiguation step. This view is inspired by the information bottleneck objective (Tishby et al., 1999):

$$\min_{p(t|x)} I(X;T) - \beta I(T;Y), \qquad (1)$$

where $I(X;T)$ penalizes retaining too much information from the input, $I(T;Y)$ rewards preserving information about the ambiguity label, and β controls this trade-off. We do not optimize a strict IB objective or estimate mutual information. Instead, the bottleneck is imposed by constraining the intervention channel.

Input and perturbation principle. As illustrated in Figure 1, our pipeline starts from an input question and applies valid perturbations under three principles. Let X be the original question, $Y \in \{0, 1\}$ the ambiguity label, and $T = E(X)$ a perturbed question. A valid intervention E should satisfy three principles. **Low capacity.** The perturbation should make only a small surface-level change, so that T remains close to X. **Label-preserving.** The perturbation should preserve the ambiguity label, so that T is ambiguous if and only if X is ambiguous. **Semantic-blind.** The perturbation may use lexical rules, surface templates, or importance scores to decide where to intervene, but it should not generate a semantic rewrite or choose among possible interpretations of the question. Together, these principles prevent the perturbation from turning ambiguity detection into ambiguity resolution by perturbation.

Perturbation pipeline. The perturbation module produces a small set of perturbed questions $\mathcal{T}_k(X) = \{T_1, \ldots, T_k\}$. In the white-box setting, model-internal signals rank input tokens or short spans, and interventions such as replacement or deletion are applied to the top-ranked positions. In the black-box setting, a constrained rule library produces surface-level perturbations without using model internals. The top-k budget controls the number of perturbation candidates used for scoring rather than defining a separate maximization objective. We describe the white-box and black-box perturbation modules in the next two subsections.

Multiple sampling and scoring. After perturbation, we sample from both the original and perturbed questions. Specifically, we draw $m = 5$ answers from X and $m = 1$ answers from each

$T_j \in \mathcal{T}_k(X)$, then combine all sampled answers into the full answer set:

$$R_X^{\mathcal{T}} = \{r_i\}_{i=1}^5 \cup \bigcup_{j=1}^{k} \{\tilde{r}_{j,i}\}_{i=1}^5. \quad (2)$$

The resulting answer set can be scored by different uncertainty estimators. In this work, we use $\mathrm{EigV(Agree)}$ from Lin et al. (2024b). And the detail technology of $\mathrm{EigV(Agree)}$ can be found in Appendix A.

3.2 White-Box Perturbation

White-box perturbation uses model access to propose low-capacity interventions that are likely to produce informative answer distortion. We focus on token because sentence-level rewrites are too high capacity and can blur detection with resolution, while token-level changes give the cleanest bottleneck test. The white-box component is not used to rewrite the question. It only identifies where a semantics-blind intervention should be applied. Attention-based scores were less reliable in preliminary experiments, so we use three attribution methods that more directly track output sensitivity. Given token scores $a_1, \ldots, a_n$, the intervention set is selected as

$$S_k(X) = \mathrm{TopK}_{i \in [n]} a_i, \qquad E_{\mathrm{wb}}(X) = E(X; S_k), \quad (3)$$

where $E(X; S_k)$ masks, deletes, or replaces only the selected tokens. The importance score only proposes E, while ambiguity is measured by the multiple-sampling distortion described above.

Vanilla Gradient. Vanilla Gradient decodes the first token greedily, $y^\star = \arg\max_y p_f(y \mid X)$, backpropagates $\log p_f(y^\star \mid X)$ to the input embeddings, and scores each token by $a_i = \|\nabla_{e_i} \log p_f(y^\star \mid X)\|_2$.

DiffMask. DiffMask learns a continuous mask over input embeddings (De Cao et al., 2020). We adapt it to find a sparse set of tokens whose masking changes the output entropy,

$$m^\star = \arg\max_{m \in [0,1]^n} |H(p_f(\cdot \mid X_m)) - H(p_f(\cdot \mid X))| - \lambda \|m\|_1. \quad (4)$$

Here m_i controls how much token i is masked, X_m is the masked input, $H(\cdot)$ is the entropy of the next-token distribution, and λ penalizes large masks so that the selected intervention remains low capacity.

Logit Lens. Logit Lens projects intermediate hidden states through the final layer normalization and unembedding matrix to obtain layerwise vocabulary distributions (nostalgebraist, 2020; Belrose et al., 2023). We use these latent predictions to identify tokens whose internal trajectory becomes strongly aligned with the final answer or with answer-relevant alternatives. These tokens are plausible places where the model commits to one interpretation, making them useful targets for low-capacity intervention.

3.3 Black-Box Perturbation

The black-box setting assumes only sample-level query access to the model. Gradients, hidden states, and token-level scores are all unavailable, so the detector can observe nothing beyond the text the model returns. Under this constraint we cannot inspect where the model allocates attention or how confident it is internally; we can only perturb the input and watch how the output distribution responds. We therefore probe the input through a fixed library of *semantics-blind* rewrites and identify informative rules as those that induce a large shift in the multi-sample output distribution. By semantics-blind we mean that each rule fires purely on surface features, namely lexical category, orthographic form, or question template, and never consults a model generated reading of the input. This property is what keeps detection separate from mitigation: a rule that does not understand the question cannot quietly resolve its ambiguity before we have had a chance to measure it.

Single-word rules. At the single-word level, we apply part-of-speech-constrained synonym substitutions for verbs, nouns, adjectives, and adverbs using WordNet (Miller, 1995). For each targeted word, we restrict substitution to synonyms drawn from its most frequent synset, filter out multi-word and rare candidates, and reinflect the chosen synonym so that its tense, number, or degree matches the original word form. This keeps each rewrite grammatical while changing only the lexical choice. We additionally include two orthographic rules that operate on the abbreviated forms of function words, namely contraction expansion and its inverse, contraction (for example *Who's* ↔ *Who is*). Each of these rules alters a single content word or its surface form while leaving the referential structure of the question, and therefore the set of interpretations it admits, intact.

Phrase- and sentence-level rules. At the phrase level, we prepend short prefixes of three registers: test-like (for example *According to the passage*), conversation-like (for example *Hey so*), and instruction-like (for example *Given the information provided*). These prefixes preserve the question stem verbatim and shift only its register, leaving the underlying query unchanged. We further apply two transformations to the leading wh word. WH expansion replaces a bare wh word with an equivalent determiner phrase (for example *Who* → *Which person*, *When* → *At what time*), and WH rephrasing substitutes the wh word with a semantically aligned alternative (for example *Who* → *What person*, *How* → *By what means*). Both wh rules are guarded so that they do not fire when the wh word is acting as a determiner, as in *What color was it*, where expansion would distort the question; in such cases, only the conservative *what/which* swap is permitted. These edits are more disruptive than single-word rewrites but still draw on fixed templates rather than free generation, which keeps them within the semantics-blind regime.

Aggregating the ambiguity signal. Given a question X, we apply every applicable rule to obtain a set of variants $\{E_1(X), \ldots, E_K(X)\}$, then draw N samples from the model for the original question and for each variant under the same QA prompt and context. Rather than scoring each rewrite in isolation, we pool the model answers from the original question and all of its variants into a single set and compute one dispersion score over that pooled set using $\mathrm{EigV(Agree)}$. The intuition is that an unambiguous question yields answers that remain stable across the entire family of surface variants, whereas an ambiguous one fractures into competing answers as different rewrites nudge the model toward different readings. Aggregating across variants rather than scoring them separately gives a single, lower variance estimate of how much the answer distribution moves under semantics-preserving perturbation, which is precisely the quantity our information bottleneck view identifies with input ambiguity.

4 Experiments

Our evaluation addresses three research questions:
• **RQ1** (§4.2). Does our perturbation-based detector separate ambiguous from unambiguous questions more reliably than existing output-side and prompt-based baselines?
• **RQ2** (§4.3). What is the relationship between different perturbations and different importance? When applied in isolation, does each single perturbation type already carry an ambiguity signal, and how consistent are different attribution methods in terms of which words they flag?
• **RQ3** (§4.4). When detector output drives downstream mitigation, does it yield higher answer correctness than generation without detection?

4.1 Experimental Setting

Models and Datasets. The main results use three base LLMs that span three model families and a 4B–27B scale range. For text QA we use Qwen3-4B-Instruct (Yang et al., 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Gemma-3-27B-it (Kamath et al., 2025). For vision QA we use the corresponding vision-language counterparts Qwen2.5-VL-3B-Instruct (Bai et al., 2025), Gemma-3-27B-it, and InternVL3-8B (Zhu et al., 2025). All experiments are run on a single NVIDIA A100 80 GB GPU.

We evaluate on four English QA benchmarks with explicit ambiguity annotations. ABG-CoQA (Guo et al., 2021) is a passage-grounded conversational QA dataset with ambiguity-annotated, from which we use a subsample of 200 items balanced between ambiguous and unambiguous. AmbigQA-w-Ctx is built from the configuration of AmbigQA (Min et al., 2020) by sampling 500 validation items, and we attach a Wikipedia passage as context for each question. AQuA (Jang and Kim, 2026) provides 720 referentially ambiguous and unambiguous visual questions over real images. VQ-FA (Chen et al., 2025a) provides visually ambiguous and unambiguous questions and we sample 720 questions as well.

Compared Methods. We compare two variants of our framework against four uncertainty-based baselines. All methods share the same base LLM and the same NLI scorer (`deberta-large-mnli` (He et al., 2020)) when an agreement matrix is needed.

Direct Prompt (Kadavath et al., 2022) asks the model in a single prompt whether the question is ambiguous, and uses $\Pr(\mathrm{YES})$ as the score. *EigV(Agree)* (Lin et al., 2024b) draws $K{=}10$ samples from the original question and reports the eigenvalue dispersion of the NLI agreement matrix. *EigV(Disagree)* (Lin et al., 2024b) reuses the

Table 1: Ambiguity-detection AUROC across three model families and four QA benchmarks. **Bold** marks the best score within each model block per column, and underline marks the second best. Vision columns use the vision-language counterpart of the text model (Qwen2.5-VL-3B for Qwen, InternVL3-8B for the Llama row, and the same Gemma-3-27B-it checkpoint for Gemma).

		Text QA		Vision QA		
Model	**Method**	**ABG-CoQA**	**AmbigQA-w-Ctx**	**AQuA**	**VQ-FA**	**Avg**
Qwen3-4B (Qwen2.5-VL-3B)	Direct Prompt (Kadavath et al., 2022)	0.542	0.525	0.359	0.525	0.488
	EigV(Agree) (Lin et al., 2024b)	0.521	0.498	0.644	0.562	0.556
	EigV(Disagree) (Lin et al., 2024b)	0.543	0.503	0.647	0.587	0.570
	Semantic Entropy (Kuhn et al., 2023)	0.493	0.491	0.674	0.477	0.534
	Ours (Black-Box)	0.615	**0.630**	**0.755**	0.563	**0.641**
	Ours (White-Box)	**0.629**	0.530	0.708	**0.613**	0.620
Llama-3.1-8B (InternVL3-8B)	Direct Prompt (Kadavath et al., 2022)	0.509	0.545	0.594	0.477	0.531
	EigV(Agree) (Lin et al., 2024b)	0.553	0.552	0.671	0.575	0.588
	EigV(Disagree) (Lin et al., 2024b)	0.565	0.519	0.586	0.582	0.563
	Semantic Entropy (Kuhn et al., 2023)	0.560	0.558	0.661	0.532	0.578
	Ours (Black-Box)	**0.634**	0.501	**0.748**	0.603	0.621
	Ours (White-Box)	0.607	**0.562**	0.712	**0.636**	**0.629**
Gemma-3-27B	Direct Prompt (Kadavath et al., 2022)	0.518	0.501	0.719	0.563	0.575
	EigV(Agree) (Lin et al., 2024b)	0.532	0.558	0.631	0.509	0.557
	EigV(Disagree) (Lin et al., 2024b)	0.509	0.527	0.551	0.517	0.526
	Semantic Entropy (Kuhn et al., 2023)	0.506	0.556	0.593	0.505	0.540
	Ours (Black-Box)	**0.650**	0.516	**0.790**	0.585	**0.635**
	Ours (White-Box)	0.580	**0.563**	0.714	**0.594**	0.613

same samples and use NLI disagreement matrix instead. *Semantic Entropy* (Kuhn et al., 2023; Farquhar et al., 2024) clusters the same samples by bidirectional NLI entailment and reports the entropy over cluster sizes.

Configurations. Every sampling-based method shares the same decoding strategy with temperature 0.5, top-p 0.9, max_new_tokens=128. Our perturbation variants draw 1 greedy sample (but less samples from the original question) from each perturbed question, with k=3 top-attributed words and up to v=4 part-of-speech-compatible WordNet variants per word. DiffMask uses 30 optimization steps with 4 greedy decode steps per step (λ=0.5). For Gemma-3-27B-it we load weights in bf16 with frozen parameters and gradient checkpointing, and the DiffMask attribution additionally runs through INT8 quantization via `bitsandbytes` to fit its autoregressive forward chain in memory. We report ROC-AUC (AUROC) for binary ambiguity classification on items where every method returns a valid score following previous works (Lin et al., 2024b).

4.2 Main Results (RQ1)

This experiment studies whether our perturbation based detector can better tell ambiguous questions from unambiguous questions. In our method, we make small, label preserving, and semantics blind changes to the input, and then check whether the answers become less stable. We do this experiment to test our idea that applying small and semantic-blind changes to the input keeps the true ambiguity signal better than just looking at the output variance. Following the setup in Section 4.1, we evaluate on four QA benchmarks across model families and compare our two variants against four uncertainty-based baselines. We show the results in Table 1. We have the following observations:

- Across all three model families, at least one variant of our method achieves the best average AUROC, and our two variants together occupy the top two average positions on every model block, with an absolute improvement of up to 0.07 in average AUROC over the strongest baseline. This shows that adding label-preserving input perturbations on top of multi-sample scoring produces a more reliable ambiguity signal.
- The improvement is largest on the two visual ambiguity benchmarks, where our black-box variant on AQuA goes up to 0.790 AUROC. Referential and visual focus ambiguity is harder to surface from output-side variance alone because the model can return a fluent answer for any chosen referent, so the gain from our perturbation is larger here.
- On the two text benchmarks, the black-box variant is generally stronger on ABG-CoQA across all

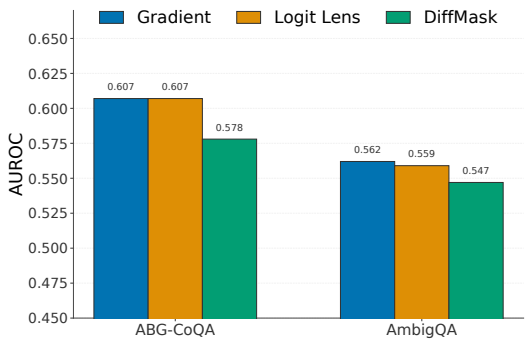


Figure 2: Performance comparing different importance methods in two datasets on Llama. The results show that though the performance of importance methods is stable, Vanilla Gradient shows a slightly better result.

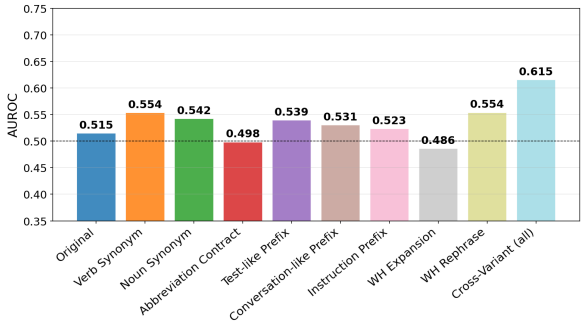


Figure 3: AUROC comparison across individual perturbation rules and their combined cross-variant setting. While several single perturbations improve over the original input, combining all perturbation variants achieves the strongest ambiguity detection performance.

three models, while the gain on AmbigQA-w-Ctx is smaller than on the other three datasets especially for white-box variant. AmbigQA-w-Ctx attaches a long Wikipedia passage to each question, so the model often has enough contextual evidence to commit to one reading regardless of how the question is perturbed. This suggests that input-side perturbation is most useful when the question itself carries the ambiguity, and that long retrieved context can partially mask the ambiguity signal by stabilizing the model's answer distribution.

4.3 Analysis of Different Importance and Perturbations (RQ2)

Choice of Importance Methods. We test whether the importance method inside the white-box detector matters for detection. Using Vanilla Gradient, Logit Lens, and DiffMask (Section 3.2) as three ways to select which tokens to perturb, we report AUROC on ABG-CoQA and AmbigQA in Figure 2. More results can be found in Appendix B.1. The results show that three methods achieve similar AUROC scores on each dataset and model, with the gap between the best and worst remaining around 0.02, and Vanilla Gradient being slightly more stable across settings. This indicates that the white-box detector does not rely on one specific importance method chosen by us, and that the ambiguity signal is mainly carried by the perturbation.

Choice of Different Perturbation Rules. Figure 3 reports detection AUROC for each black-box rule individually, alongside the cross-variant setting that pools the answers from the original question and all rules into a single dispersion score. Cross-variant settings achieve the best performance. This confirms that the ambiguity signal is distributed across heterogeneous perturbations rather than concentrated in any one, and that aggregating the answer distribution over the full family of rules is what makes detection effective.

4.4 Mitigation Performance (RQ3)

Detection is valuable only to the extent that it improves downstream answers. We therefore evaluate whether driving mitigation with the detector's output yields higher answer correctness than generation without detection. To this end we introduce *UQ-Augmented Re-asking* (UAR), which reuses the perturbation and uncertainty artifacts produced during detection and spends a single additional forward pass to recover a more reliable answer.

Method. For each question, the detector has already sampled the model under the original phrasing and every rewrite rule. UAR ranks rules by the dispersion they induce relative to the original, $\delta(r) = U(\text{rule } r) - U(\text{ORIGINAL})$, keeps the top k with positive δ (default $k = 3$), and for each extracts the rewritten question and its modal answer. These (rewrite, answer) pairs are inserted into the prompt as statements of the form "when rephrased as X, the model answers Y," with an instruction to weigh the passage and commit to a single answer. A question for which no rule has a positive δ gets the unchanged baseline prompt, so re-asking fires only when the detector has localized interpretation-sensitive behavior.

Setup and results. We compare against *Original*, the model's first answer to the question with no uncertainty information; both conditions share the same model, context, and decoding setup (greedy, single sample), so any difference comes from the

Table 2: UAR vs. the original baseline on Qwen3-4B, greedy decoding. UAR re-prompts the model with detector-derived evidence; Original is the model's first answer to the unperturbed question.

Config	Group	Original	UAR	Δacc
AmbigQA-w-Ctx	ambiguous	12.19	**26.52**	+14.33
	non_ambig.	16.28	**34.35**	+18.07
ABG-CoQA	ambiguous	22.76	**34.15**	+11.39
	non_ambig.	17.06	**25.21**	+8.15

prompt. As Table 2 shows, UAR improves Exact Match on every subset of both datasets, by +8.15 to +18.07 points. The gain also appears on the gold non-ambiguous subset because re-asking fires on questions whose *model* answers disagree across rewrites, a broader signal than the human label: many gold non-ambiguous questions still destabilize the model, and the injected evidence helps it commit to a single answer.

4.5 Ablation Study

Influence of Semantic-blind Perturbation This experiment tests whether the semantic-blind constraint on the perturbation step is actually necessary. As discussed in Section 3.1, our framework requires the perturbation to be low-capacity and semantics-blind, so that the input is rewritten without choosing among its possible readings. A natural alternative is to let an LLM directly paraphrase the question, which is a high-capacity transformation that may resolve the ambiguity during rewriting. We report both the absolute uncertainty value and the AUROC in Figure 4 using Qwen2.5-VL-3B and AquA. The results show that replacing our semantic-blind perturbation with LLM paraphrasing lowers the absolute uncertainty value to even below the EigV(Agree) baseline that uses no input perturbation, and the AUROC also drops from 0.708 to 0.637. This shows that LLM paraphrasing does not just fail to add a useful signal, it can potentially remove ambiguity that the baseline could still pick up from sampling alone.

Rule Execution: Ours vs. LLM-based. In the black-box setting, a fixed rule can be executed either deterministically, with WordNet synonyms and string templates, and the rewrite is strictly semantics-blind, or by handing the same rule to an LLM as an instruction. Unlike free LLM paraphrasing, the rule is fixed in advance; the question is whether this constraint is enough to stop the LLM from understanding, and thereby disambiguating,

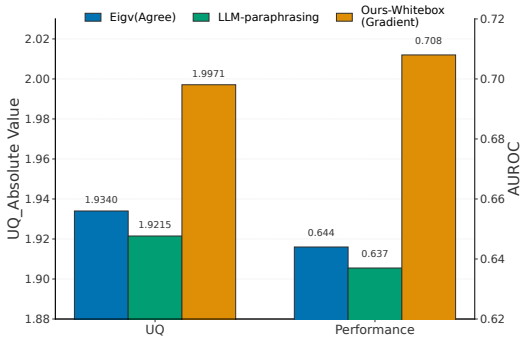


Figure 4: Comparison of performance and absolute value of uncertainty on ambiguous questions from AquA. The results show that directly using LLM-paraphrasing can actually reduce the uncertainty.

Table 3: Detection AUROC for two realizations of the same rule set. Our perturbations outperform LLM-based ones, while remaining semantics-blind and cheaper.

Model	Dataset	Ours	LLM
Qwen3-4B	ABG-CoQA	**0.615**	0.598
	AmbigQA	**0.630**	0.617
Qwen2.5-VL-3B	AQuA	**0.755**	0.715
	VQ-FA	**0.563**	0.538

the original question. The two realizations share an identical rule inventory and differ only in the capacity of the mechanism that executes them. As shown in Table 3, across four dataset settings spanning text (Qwen3-4B) and vision (Qwen2.5-VL-3B), ours perform better than the LLM-based one. It shows that even a tightly constrained rule, once routed through a model that must understand the input, erodes rather than preserves the ambiguity signal, which supports our use of semantics-blind rewrites for detection.

5 Conclusion

This work studies ambiguity detection as a prerequisite for reliable question answering. We propose a perturbation-based framework that preserves ambiguity signals by avoiding semantic rephrasing, and instantiate it with both white-box and black-box detectors. Experiments across open-domain, conversational, and visual QA datasets show that our method more effectively distinguishes ambiguous from unambiguous questions than output-side sampling and paraphrase-based baselines. When its output drives evidence-augmented re-asking, the detector further improves downstream answer accuracy on under-specified questions.

Limitations

First, the black-box detector relies on a manually designed rule library built for English, so applying it to other languages requires reconstructing these rules. Second, the white-box detector depends on access to model internals, which is unavailable in many deployment settings such as closed-source APIs.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Chongyan Chen, Yu-Yun Tseng, Zhuoheng Li, Anush Venkatesh, and Danna Gurari. 2025a. Acknowledging focus ambiguity in visual questions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1228–1238.

Tiejin Chen, Kuan-Ru Liou, Mithun Shivakoti, Aaryan Gaur, Pragya Kumari, Meiqi Guo, and Hua Wei. 2025b. Abg-SciQA: A dataset for understanding and resolving ambiguity in scientific questions. In *ICLR 2025 Workshop on Data Problems*.

Tiejin Chen, Huaiyuan Yao, Jia Chen, Evangelos E Papalexakis, and Hua Wei. 2026. Every response counts: Quantifying uncertainty of llm-based multi-agent systems through tensor decomposition. *arXiv preprint arXiv:2604.08708*.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore. Association for Computational Linguistics.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, Online. Association for Computational Linguistics.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6970–6984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. 2024. SPUQ: Perturbation-based uncertainty quantification for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346, St. Julian's, Malta. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-CoQA: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 19023–19042. PMLR.

Jihyoung Jang and Hyounghun Kim. 2026. AQuA: Toward strategic response generation for ambiguous visual questions. In *The Fourteenth International Conference on Learning Representations*.

Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. 2025. Teaching vision-language models to ask: Resolving ambiguity in visual questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3619–3638, Vienna, Austria. Association for Computational Linguistics.

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

Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. 2024. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754.

Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, and 191 others. 2025. Gemma 3 technical report. *ArXiv*, abs/2503.19786.

Kimiya Keyvan and Jimmy Xiangji Huang. 2022. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys*, 55(6):1–40.

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sanggoo Lee, and Taeuk Kim. 2024. Aligning language models to explicitly handle ambiguity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1989–2007.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. CLAM: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*.

Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joonsuk Park, Sang-Woo Lee, and Kyomin Jung. 2023. Asking clarification questions to handle ambiguity in open-domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11526–11544, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024a. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10351–10368, Miami, Florida, USA. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024b. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 790–807.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *KDD 2025 - Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 2 of *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 6107–6117. Association for Computing Machinery.

Yaokun Liu, Yifan Liu, Phoebe Mbuvi, Zelin Li, Ruichen Yao, Gawon Lim, and Dong Wang. 2026. Mind the ambiguity: Aleatoric uncertainty quantification in llms for safe medical question answering. In *Proceedings of the ACM Web Conference 2026*, pages 9373–9384.

George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *Advances in Neural Information Processing Systems*, volume 37.

nostalgebraist. 2020. Interpreting GPT: The logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`. LessWrong post.

Jie Ren, Yao Zhao, Tu Vu, Peter J. Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models. In *Proceedings on I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models at NeurIPS*

2023 Workshops, volume 239 of *Proceedings of Machine Learning Research*, pages 49–64. PMLR.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Naftali Tishby, Fernando C. Pereira, and William Bialek. 1999. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377.

Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. 2025. Interactive agents to overcome ambiguity in software engineering. *arXiv preprint arXiv:2502.13069*.

Adam Yang, Chen Chen, and Konstantinos Pitas. 2024. Just rephrase it! uncertainty estimation in closed-source language models via multiple rephrased queries. *arXiv preprint arXiv:2405.13907*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Michael Zhang, W Bradley Knox, and Eunsol Choi. 2025. Modeling future conversation turns to teach llms to ask clarifying questions. In *International Conference on Learning Representations*, volume 2025, pages 60722–60742.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. 2024. CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10746–10766, Bangkok, Thailand. Association for Computational Linguistics.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Detail Technology of Uncertainty Estimation

We first use an NLI model to compute a pairwise semantic agreement matrix A over answers in $R_X^{\mathcal{T}}$. For answers r_i and r_j, we set

$$A_{ij} = \frac{1}{2}\left(p_{\text{ent}}(r_i \Rightarrow r_j) + p_{\text{ent}}(r_j \Rightarrow r_i)\right), \quad (5)$$

where p_{ent} is the entailment probability assigned by the NLI model, with $A_{ii} = 1$. We then build the degree matrix D, where $D_{ii} = \sum_j A_{ij}$, and compute the symmetric normalized graph Laplacian

$$L = I - D^{-1/2} A D^{-1/2}. \quad (6)$$

Let $\lambda_1, \ldots, \lambda_N$ be the eigenvalues of L, where $N = |R_X^{\mathcal{T}}|$. Following the spectral clustering motivation in Generating with Confidence, smaller eigenvalues indicate semantic clusters among sampled answers. We compute

$$U(X) = \text{EigV}(\text{Agree}(R_X^{\mathcal{T}})) = \sum_{q=1}^{N} \max(0, 1-\lambda_q). \quad (7)$$

A larger value indicates that the combined original and perturbed samples separate into more semantic modes, which is the signal we use for ambiguity.

B More Experiments

B.1 More Results on Different White-box Importance

We test whether the importance method inside the white-box detector matters for detection using all models we used for text-only datasets and the full results can be found at Figure 6. The conclusion is consistent with the main text that our choice of importance is stable and Vanilla Gradient is slightly better.

B.2 More Ablation

More White-box Importance To test whether our choice of white-box importance makes sense, we compare with our white-box methods with more white-box importance methods:

• attention: recursively multiplies per-layer attention matrices to compute the total information flow from each input token to the output.

• predictive entropy: assigns each input token an importance score equal to the entropy of the model's next-token distribution at that position.

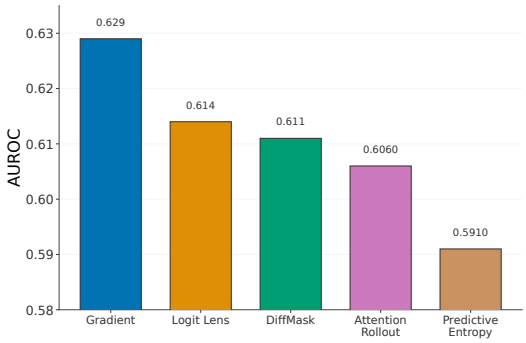


Figure 5: Performance comparing more white-box importance methods on Abg-CoQA and Qwen3. The results show that our choice is better than simple attention or predictive entropy.

We show the results in Figure 5, which shows that simply using attention or entropy cannot get better results, and our choice of importance methods is reasonable.

C Detail Methods for Black-Box Perturbation

This appendix lists the full set of rules used by the black-box perturbation module described in Section 3.3. All rules are semantics-blind, meaning that each rule fires on surface features such as part-of-speech tag, orthographic form, or the leading wh word, and never consults a model-generated reading of the input. The rules are organized into two groups according to the granularity of the edit.

C.1 Group A: Single Word Level

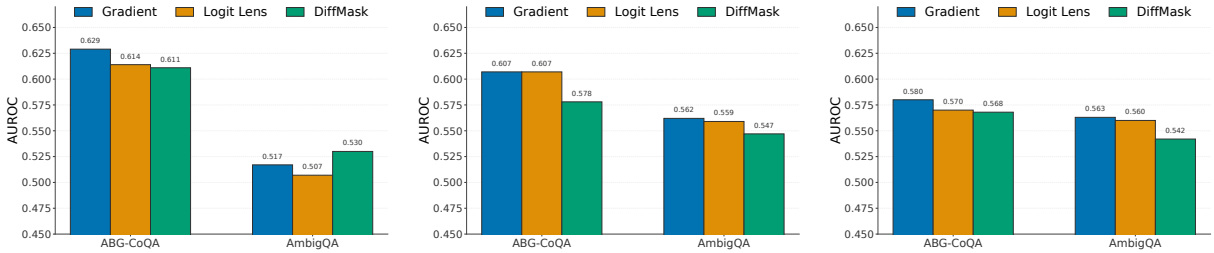
Rules A1 to A4 substitute a single content word with a WordNet (Miller, 1995) synonym drawn from the first synset of the target word, with multi-word, very short, length-mismatched, and rare candidates filtered out. The selected synonym is then reinflected to match the tense, number, or degree of the original word.

• **A1: Verb Synonym.** Replace a verb with a WordNet synonym, reinflected to match the original verb form. Example: *Who won the game?* → *Who prevailed the game?*

• **A2: Noun Synonym.** Replace a noun with a WordNet synonym, with number preserved. Example: *When was the building completed?* → *When was the structure completed?*

• **A3: Adjective Synonym.** Replace an adjective with a WordNet synonym, with degree preserved. Example: *What is the largest city?* → *What is the biggest city?*

• **A4: Adverb Synonym.** Replace an adverb with



(a) Qwen3. (b) Llama3.1. (c) Gemma3-27b.

Figure 6: Performance comparing different white-box importance methods in text-only datasets. The results show that though the performance of importance methods is stable, Vanilla Gradient shows a slightly better result.

a WordNet synonym. Example: *How quickly does it grow?* → *How rapidly does it grow?*

• **A5: Abbreviation Expand.** Expand a contraction into its full form. Example: *Who's the author?* → *Who is the author?*

• **A6: Abbreviation Contract.** The inverse of A5, contracting a full function-word sequence into its abbreviated form. Example: *Who is the author?* → *Who's the author?*

C.2 Group B: Sentence and Phrase Level

Rules B1 to B3 keep the question stem verbatim and only change its leading register through a short prefix. Rules B4 and B5 act on the wh word at the beginning of the question through a small fixed mapping table, and both are guarded so that they do not fire when the wh word is acting as a determiner, as in *What color was it*.

• **B1: Test-like Prefix.** Prepend a short test-style phrase. Examples include *According to the passage*, *Based on the text above*, and *As stated in the document*.

• **B2: Conversation-like Prefix.** Prepend a short conversational opener. Examples include *Hey so*, *Quick question*, and *By the way*.

• **B3: Instruction Prefix.** Prepend a short instruction-style phrase. Examples include *Please answer the following question*, *Your task is to answer*, and *Respond to this question*.

• **B4: WH Expansion.** Replace a bare wh word with an equivalent determiner phrase. Example: *Who won?* → *Which person won?*

• **B5: WH Rephrase.** Replace the wh word with a semantically aligned alternative. Example: *How did it happen?* → *By what means did it happen?*