RULEUSION: GENERATOR-FREE METAMORPHIC TESTING FOR REFERENCE-FREE HALLUCINATION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Without a reference answer or access to a model's internals, a hallucination has to be detected from the model's own replies. Sampling detectors ask the same question many times and read disagreement as a warning. When every sample is the same, the agreement can mean the model is right or that it repeats one error; we call this *decoding collapse*, and these detectors then have little to measure. Detectors that change the input instead, such as SAC^3 and MetaQA, pay a large language model (LLM) to write the tests. We present Ruleusion, which checks an answer by asking questions whose answers follow from it by logic. Each test is either a fixed rewrite of the question or a short follow-up about the model's own answer. The rewrite states a logical relation, such as negation, presupposition, or entailment, so the expected answer is known before the model is queried, and the reply is graded locally. No generative model writes or grades any test. On answers that five LLMs gave to questions from five short-answer datasets, Ruleusion achieves the highest average AUROC of seven detectors, 0.744, ahead of the strongest baseline, SAC^3 (0.722), and it is the most accurate detector at every call budget: with 6.5 calls per question it already matches SAC^3 at 27 calls. It uses 54% fewer calls and 70% fewer tokens than SAC^3. On the half of answers where all twelve samples coincide, every other detector loses accuracy, SelfCheckGPT and Semantic Entropy falling to 0.53–0.56 AUROC, while Ruleusion keeps 0.69, the same as on the other half.

1 INTRODUCTION

Large language models answer factual questions in fluent sentences, and some of those answers are wrong. We call such an answer a hallucination. It reads as fluently as a correct answer, so a user who does not know the fact cannot tell the two apart. This paper studies how to detect it: given a question and the answer a model returned, the detector outputs a score for how likely the answer is a hallucination. The model is reached only through its API: the detector can ask it questions and read its replies, but it has no reference answer and no view of the model's internals. We study short answers to factual questions, from a single word to one sentence.

Existing detectors have two problems in this setting: they miss errors, and they are expensive. Fact-checkers such as FActScore and FacTool can check only facts that their reference source covers (Min et al., 2023; Chern et al., 2023), and white-box detectors need internals that an API does not expose (Chen et al., 2024a; Sriramanan et al., 2024). SelfCheckGPT (Manakul et al., 2023) and Semantic Entropy (Kuhn et al., 2023; Farquhar et al., 2024) use only the model's replies: they ask the same question many times, twelve in our experiments, and read disagreement as a sign of hallucination. When every sample is the same, there is no disagreement to read, although the repeated answer may be a confident error. We call this *decoding collapse*, and it is where these detectors miss errors: it occurs on half of our answers, and there SelfCheckGPT and Semantic Entropy fall to 0.53–0.56 AUROC, barely above chance; Figure 1 shows one case. MetaQA (Yang et al., 2025) observed collapse in SelfCheckGPT's samples, and SAC^3 (Zhang et al., 2023) likewise shows that self-consistency checks miss answers a model repeats consistently.

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

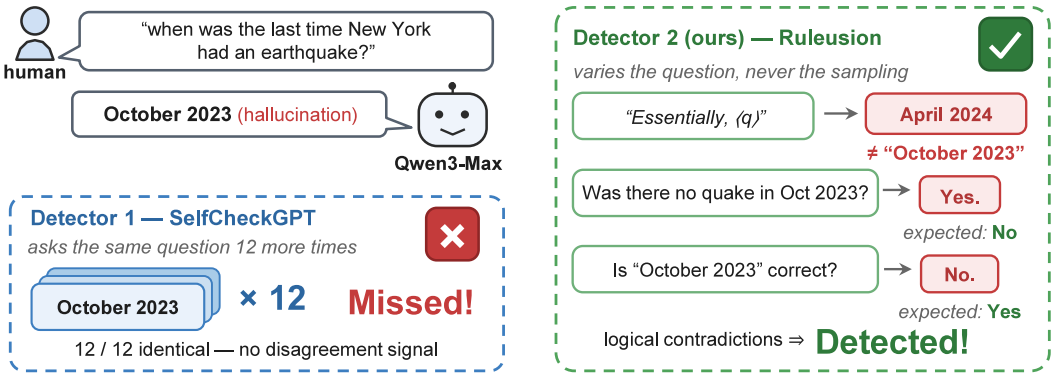


Figure 1: A Natural Questions example of decoding collapse: all 12 samples repeat the same wrong answer. One fixed rewrite, and follow-up questions about that answer, produce replies that cannot all be right.

SAC^3 and MetaQA change the input instead of resampling, but they pay an LLM to write the tests, and this is where the cost comes from: SAC^3 spends 27 calls per answer, 19 of them writing and grading, and MetaQA 20 calls, 8 of them writing. The LLM that writes the tests can itself hallucinate or drift from the question, so part of SAC^3's writing calls go to checking that its paraphrases kept the meaning. In short, these detectors depend on an LLM, and its frequent calls, compounded by the imprecision of the tests it writes, drive up their cost.

Ruleusion rests on a simple observation: if an answer is right, questions logically tied to it must get answers that fit it. Hand-written rules rewrite the question. The rule states a relation to the original, such as negation, presupposition, or entailment, and that relation fixes the expected answer once the original answer is known. The other tests are follow-up questions about the original answer: if that answer were right, what else would have to be true? Every test therefore has a known expected reply before the target is queried, and the reply is graded locally, by its yes/no polarity or by natural language inference (NLI), which judges whether two statements contradict each other. Logic, not an LLM, fixes each expected reply, so the tests are precise and still work when all samples agree. Rules write and check the tests and the replies are graded locally, so only the test replies need LLM calls. Ruleusion therefore addresses both problems above.

We test Ruleusion on 5,995 answers that five LLMs gave to 1,199 questions from five question-answering (QA) datasets, against six baselines that also see only the model's replies. The datasets' reference answers are used only to score the detectors, which never see them.

The main contributions are:

- We measure decoding collapse: on half of our answers all twelve samples coincide, and there SelfCheckGPT and Semantic Entropy lose about 11 AUROC points, falling to 0.53–0.56. Ruleusion, whose rules fix the expected reply of every test in advance, keeps 0.69, as on the other half, the only detector that keeps its accuracy there.

- We propose Ruleusion: hand-written rewrites state a logical relation to the question, including negation, presupposition, and entailment, so the expected answer is fixed before the model is queried. Follow-up probes do the same for the original answer. The score of a question uses only that question's violations: average inside each group of tests, then across groups.

- On 25 model–dataset cells, **Ruleusion attains the highest average AUROC of seven detectors, 0.744, and the best accuracy at every call budget.** Given at least as many calls, SAC^3 trails it by 4.2–4.5 points and MetaQA by 3.2–3.6 points. It uses 54% fewer calls than SAC^3 and 37% fewer than MetaQA.

2 Related Work

Sampling and uncertainty. SelfCheckGPT measures inconsistency between a response and stochastic samples (Manakul et al., 2023). Semantic Entropy clusters samples by meaning and computes entropy over those clusters (Kuhn et al., 2023; Farquhar et al., 2024). Semantic Volume instead measures how spread out the embeddings of the samples are (Li et al., 2026). All three read variation across draws of the same question, so they become inaccurate exactly when the model repeats itself: under decoding collapse none is left. A test whose expected reply is fixed before the query does not need it, because a repeated wrong answer must still match that reply.

Perturbation and metamorphic testing. Metamorphic testing checks relations between the outputs of related inputs (Chen et al., 2018; Segura et al., 2016), and CheckList brings behavioral tests of this kind to NLP (Ribeiro et al., 2020). The detectors closest to ours change the input, but they are expensive and their tests imprecise: an LLM writes the tests, at the cost of extra calls, and the tests vary with the model that writes them and can carry its hallucinations. SAC^3 checks whether the answers to semantically equivalent questions agree, and a cross-model variant also queries a second model (Zhang et al., 2023). MetaQA has the model write synonym and antonym mutations of its answer and verify each one (Yang et al., 2025). InterrogateLLM asks the model to reconstruct the question from its answer several times and compares the reconstructions with the question by embedding similarity (Yehuda et al., 2024). Belief Tree Propagation (BTProp) has an LLM expand a statement into a tree of logically related statements and combines the model's beliefs about them (Hou et al., 2025). A paraphrase keeps the expected answer, so a model that repeats one wrong answer can pass every paraphrase. DrHall requires relations on both the question and the answer, plus a vote over several repairs, and some of its relations add outside evidence (Wu et al., 2025). Drowzee builds both the question and the true answer from a Wikipedia knowledge base (Li et al., 2024). Both need resources outside the model. SECA uses rewrites to attack a model: it searches for rewrites that keep the meaning and still elicit hallucinations, showing that such rewrites can expose unstable answers (Liang et al., 2025). When a stated relation decides what a test expects, the test needs no call to write or check it and cannot carry an LLM's hallucination; and since the relation can demand a reply other than the original answer, repeating that answer fails such a test.

Self-evaluation and model access. Other detectors have the model write its own checks, bring in a second model or outside evidence, or read the model's internals; each is expensive or inaccurate, or needs what an API does not give. Chain-of-Verification asks the model to write questions and then revise its answer (Dhuliawala et al., 2024). LM vs. LM (LMvLM) has a second LLM examine the first over several turns (Cohen et al., 2023). Asking a model about its own citations can expose invented references (Agrawal et al., 2024). Using an LLM as a judge adds another model and that model's biases (Zheng et al., 2023). FActScore and FacTool check claims against retrieved evidence (Min et al., 2023; Chern et al., 2023). White-box detectors read hidden states, attention, or the geometry of embeddings (Chen et al., 2024a; Sriramanan et al., 2024; Chuang et al., 2024), and some pipelines add tools and cover multimodal models (Chen et al., 2024b). Checks derived by logic need none of these, because a model that answers a claim and its negation alike contradicts itself, whatever the true answer.

3 Ruleusion

Overview. Ruleusion asks new questions with fixed rules, choosing questions whose correct answers follow from the model's own answer by logic. In the example of Figure 2, the model answers "When did X happen?" with "July 2024". The rules rewrite the question into "Did X happen?", which presupposes the event and should get yes, and "In what year did X happen?", which should get 2024 (step 1). Probes then ask about the answer itself, for instance whether "July 2024" is the right answer, which should get yes (step 2). The mismatches are averaged by group into a risk $S(q)$ (step 3). Algorithm 1 gives the full procedure, and Table 1 lists the rules and probes.

What each part is for. The core of Ruleusion is that logic tells us, before we ask, what the model must reply if its answer is right. Each reply is then checked on its own, so the check works when all samples agree, and no LLM writes or grades the tests, so the only calls are those that ask them. Steps 1 and 2 supply the evidence. The Logic rewrites change the question by a relation, so

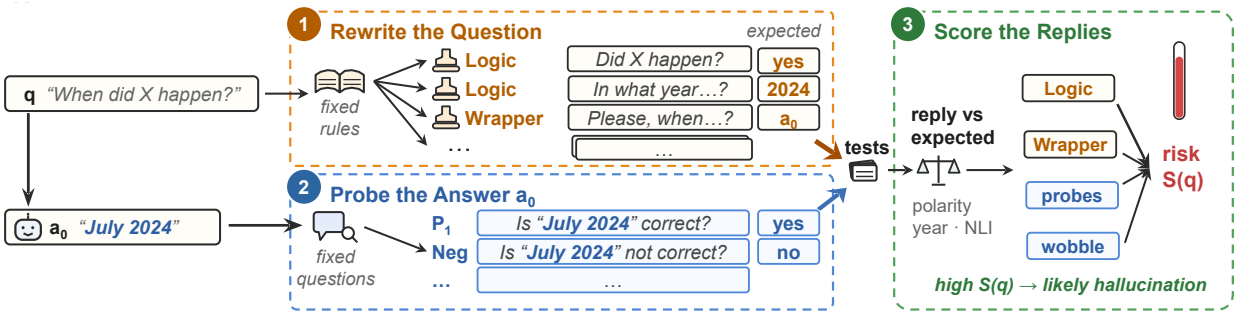


Figure 2: Overview of Ruleusion, with the running example q = "When did X happen?" and a_0 = "July 2024". The target M first answers q with a_0. Step 1: hand-written rules rewrite q. A LOGIC rule applies a named relation, such as presupposition ("Did X happen?", expected yes); a WRAPPER rule changes only the wording (expected: the same answer as a_0). Step 2: probes ask about a_0. P1 asks whether a_0 is right (expected yes), and a set chosen by the answer type follows; probes on answers that are not yes/no also return a confidence from 0 to 100, and wobble is how close that confidence is to 50. Step 3: M answers every test, each reply is scored against its expected answer, and the scores are averaged inside the groups (LOGIC, WRAPPER, probes, wobble) and then across groups into $S(q)$. A high $S(q)$ flags a_0 as a likely hallucination.

the expected reply can differ from the original answer; the WRAPPER rewrites test cheaply whether new wording moves the answer. The probes reach every question, including those no LOGIC rule fits, and pair a claim with its negation, so answering both alike is a contradiction; on answers that are not yes/no, their confidence can also reveal hesitation that identical samples hide. Step 3 grades every reply locally, so it spends no call, and averages inside each group, so the many WRAPPER tests do not outvote the rest.

Notation. For a question q, the black-box target M returns an answer a_0. Every later test is a prompt paired with its expected reply, fixed from a_0 and a stated relation before M is called again. The output is a risk $S(q) \in [0, 1]$, near 0 when the replies meet the stated expectations and near 1 when they break them.

Declared relations. Every rewrite is a hand-written function of the question text. The same question therefore produces the same tests, and we never call a model to write them. Let $\mathrm{rel}(v)$ be the relation that the rule states between a variant v and q. Once a_0 is known, that relation says what v should be answered with. No test asks an open question. Table 1 splits the rules into two layers. LOGIC rules change the question according to a named relation, such as equivalence or negation, and can therefore change the expected answer: under negation it flips. WRAPPER rules keep the meaning and change only the wording; their relation is equivalence. Rewording alone can change a language model's answer (Elazar et al., 2021), so these rewrites are tests too. A parser looks at the shape of q — for example, a yes/no question, a "when" question, or a question with "and" — and keeps only the rules that match that shape. We then keep a set $V(q)$ of at most B variants. LOGIC variants are kept first, and each WRAPPER rule contributes at most W. We set B and W in §4. A variant that fails the check is never sent to M. We also drop nested double negation inside a clause, because LLMs handle it unreliably (Yang et al., 2025; Varshney et al., 2025).

Violation scores. Let $a_v = M(v)$. The violation $s_v \in [0, 1]$ is 0 when a_v matches the answer required by $\mathrm{rel}(v)$, and 1 when it breaks that requirement. The check depends on the relation. For a negated variant, an equivalent yes/no variant, or an alternative variant, we compare yes/no polarity with the expected polarity. On a negated variant, a mismatch means a_v has the same polarity as a_0: the model failed to flip its answer. An "entailed if yes" variant scores 1 when a_0 is yes and a_v is no, and is dropped when a_0 is no, because the entailment does not apply. An "agree if and only if yes" variant contributes one score for the pair of answers: the two answers agree exactly when they should, namely when a_0 is yes. A presupposition variant expects yes, unless a_0 itself denies the presupposition, and again we score the yes/no mismatch. A variant that asks for the year, rather than

Table 1: Rule library and probe forms. A LOGIC rule rewrites the question by a named logical relation. A WRAPPER rule keeps the meaning and changes only the wording. The Bool route is for yes/no answers and the short route for every other answer. The distractor d is another candidate answer of the same kind. Parentheses give the expected replies if a_0 is right. In the patterns, S is a subject and VP a verb phrase.

Transformation rules (family / relation / operation)		
LOGIC layer		
QUAD-NEG	equivalent	Four stacked negations over the parsed clause P ($\neg\neg\neg\neg P \equiv P$): "Is it not untrue that it would not be incorrect to say that P?"
SIMPLE-NEG	negated	Single negation of the clause; the expected polarity is the opposite of a_0.
CONVERSE	equivalent	Converse relation with swapped arguments: "Is X taller than Y?" $\rightarrow$ "Is Y shorter than X?"
CONJ-ELIM	entailed if yes	"Are X and Y both P?" $\rightarrow$ "Is X P?", "Is Y P?"; each conjunct must be yes when a_0 is yes.
SAME-REL-SPLIT	agree iff yes	"Were X and Y of the same N?" $\rightarrow$ "What is the N of X?", "... of Y?"; the two answers agree iff a_0 is yes.
ALT-DECOMP	alternative	"Is X P or Q?" $\rightarrow$ "Is X P?", "Is X Q?"; expected polarity follows the alternative named in a_0.
WH-PRESUP	presupposition	"When did S VP?" $\rightarrow$ "Did S VP?"; expected yes unless a_0 denies the presupposition.
WH-YEAR	same year	"When ...?" $\rightarrow$ "In what year ...?"; the years named must agree.
WH-PASSIVE, WH-SUPERL	equivalent	Passive wording ("Who wrote X?" $\rightarrow$ "Who was X written by?"); split a superlative ("the ADJ-est N" $\rightarrow$ "the N that is ADJ-er than any other").
PAR-SWAP	equivalent	Swap two coordinated names in a yes/no question (commutativity): "Are X and Y both P?" $\rightarrow$ "Are Y and X both P?"
WRAPPER layer (all equivalent)		
TIME-FRONT, WORD-ORDER	equivalent	Move a phrase to the front: "...in 2019?" $\rightarrow$ "In 2019, ...?"
ABSTRACT	equivalent	A short frame around the unchanged question, such as "In general terms, ...".
POLITE, ADVERB, REDUNDANT	equivalent	Prefixes that do not change what would make the answer true: "Could you tell me, ..."; "In essence, ...".
Probes about the original answer (probe / which answers / what we ask)		
P1	all	Ask whether a_0 is the right answer (yes)
NEG	Bool	Ask whether the opposite yes/no value is right (no)
EXCL-ALT	Bool	Ask whether the opposite yes/no value could be right instead of a_0 (no)
NEG	short	Ask whether a_0 is not the right answer (no)
DEMORGAN	short	Ask whether a_0 and d are both wrong (no)
DISJ-INTRO	short	Ask whether at least one of a_0 and d is right (yes)
NO-OTHER	short	Ask whether a_0 is the only answer (yes)

the full time, scores 1 when the years named in a_v and a_0 do not overlap, and is dropped when either answer names no year. Equivalent variants that are not yes/no questions need a different check. Bare answers such as "2023" and "2024" are fragments, and an NLI model may call them neutral. We therefore wrap each answer as a full sentence, $t(q, a) =$ "Question: q Answer: a", and set

$$s_v = \tfrac{1}{2}\big[P_{\text{con}}\big(t(q, a_0), t(v, a_v)\big) + P_{\text{con}}\big(t(v, a_v), t(q, a_0)\big)\big], \tag{1}$$

the mean of the two contradiction probabilities. The encoder is DeBERTa-v3-large fine-tuned on MNLI, the same use of NLI as in SelfCheckGPT-NLI. The factor one-half averages the two directions. A reply that cannot be parsed for the required relation contributes no evidence.

Probes about the original answer. Every question gets one direct check, P1: we ask whether a_0 is the right answer, and the expected reply is yes. We then add one set of probes from Table 1,

Algorithm 1 Ruleusion(q, M). ROUTE keeps rules that match the shape of q. BANK is the probe set for that kind of answer. B, W, and α are set in §4.

$a_0 \leftarrow M(q)$
$V \leftarrow$ VALIDATE(ROUTE(q)); keep $\leq B$ variants, LOGIC first, $\leq W$ per WRAPPER rule
for $v \in V$ **do** $a_v \leftarrow M(v)$; $s_v \leftarrow$ VIOLATION$\big(a_v,$ EXPECTED$(\mathrm{rel}(v), a_0)\big)$
end for
$P \leftarrow \{\mathrm{P1}\} \cup$ BANK$(\mathrm{type}(a_0))$
for $p \in P$ **do**
    $(z_p, c_p) \leftarrow M(p)$; $e_p \leftarrow$ EXPECTED(p, a_0)
    $d_p \leftarrow$ MISMATCH(z_p, e_p) if p is Boolean, else equation 2
    $w_p \leftarrow 1 - |c_p - 50|/50$ if c_p is returned
end for
families $\leftarrow \{$LOGIC, WRAPPER, PROBE, WOBBLE$\}$, drop empty
return $S(q)$ by equation 3

chosen by the form of a_0. If a_0 is yes or no, the set has two yes/no probes, and both expect no. Otherwise the set has four probes. Two expect yes and two expect no, so answering yes to all of them, or no to all of them, fails the set. P1 and NEG ask a claim and its negation, as do DEMORGAN and DISJ-INTRO, so a model that answers both probes of a pair the same way contradicts itself, whatever the true answer. Let z_p be the reply and e_p the expected reply. Yes/no probes, including P1, score 1 when the polarity disagrees and 0 when it agrees. A probe in this set also asks the model to state a confidence c_p from 0 to 100. Its violation is

$$d_p = \begin{cases} c_p/100, & z_p \neq e_p, \\ (100 - c_p)/100, & z_p = e_p. \end{cases} \tag{2}$$

Thus d_p is high when the model is confident and wrong, and also high when it is right but unconfident. The same reply yields a second number, $w_p = 1 - |c_p - 50|/50$. We call it wobble. The 50 is the middle of the 0–100 scale. Wobble is 1 when the stated confidence is 50, and 0 when it is 0 or 100. d_p records whether the yes/no answer matches e_p. Wobble records whether that confidence sits at an end of the scale. Both use the reply already returned, so wobble adds no call. A reply we cannot parse is omitted. The uniqueness probe, NO-OTHER, assumes the benchmark has a single correct answer.

Scoring head. The groups above are the families in the formula. Each violation is tagged with a family f: LOGIC rewrites, WRAPPER rewrites, probes, or wobble. Let $\overline{s}_f$ be the mean inside a family that is not empty. The risk is

$$S(q) = \frac{\sum_f w_f \, \overline{s}_f}{\sum_f w_f}, \qquad w_f = \begin{cases} 1, & |f| \geq 2, \\ \alpha, & |f| = 1, \end{cases} \tag{3}$$

$S(q)$ goes to 0 when the replies meet the stated expectations, and to 1 when they break them. We average inside each family before averaging across families, so a family with many tests does not outvote a family with few. The weight α is set in §4. Equation 3 uses only the violations of this question q.

4 EXPERIMENTAL SETUP

Datasets and models. We evaluate FreshQA (Vu et al., 2024), Natural Questions (NQ) (Kwiatkowski et al., 2019), Mintaka (Sen et al., 2022), StrategyQA (Geva et al., 2021), and TruthfulQA (Lin et al., 2022). FreshQA is scored against the gold aliases of that release. In that order, they stress facts that change, short open-domain facts, questions with types and several hops, yes/no questions that need an implicit step, and questions built around common misconceptions. Each model–dataset cell contains 99, 100, 300, 500, and 200 questions. Over five models this gives 5,995 answers to 1,199 questions. The Mintaka cells are balanced over its nine complexity types, 33 or 34 questions each, so that §5 can report each type separately. The targets are DeepSeek-Chat, GPT-4o-mini, Qwen3-Max, Llama-3.3-70B, and Mistral-Large-2512. We set the variant budget to B=8 and

Table 2: AUROC averaged over five models; boldface marks the best in each column. Calls are extra LLM calls per question; Write/grade is the part of them that writes or grades tests (Table 6). Semantic Volume reuses Semantic Entropy's samples. †The paired gain of Ruleusion over 25 cells is significant (Wilcoxon, Holm-corrected over six baselines, $p < 0.05$; Table 5). Per-cell values are in Table 7.

Method	FreshQA	NQ	Mintaka	StrategyQA	TruthfulQA	Avg.	Calls	Write/grade
Ruleusion	**.783**	.760	**.800**	**.690**	.685	**.744**	12.5	0
SAC^3	.748	**.772**	.792	.672	.626	.722	26.9	18.9
MetaQA†	.752	.670	.721	.654	**.739**	.707	20.0	8.0
SelfCheckGPT†	.736	.663	.743	.628	.584	.671	12.0	0
Semantic Entropy†	.729	.681	.710	.569	.579	.654	12.0	0
Semantic Volume†	.671	.694	.717	.604	.574	.652	12.0	0
LLM judge†	.664	.625	.640	.585	.618	.626	1.0	1.0

the cap on each WRAPPER rule to $W=2$. Together with P1 and either two or four further probes, a question receives 11 or 13 tests after the shared original answer. We set $\alpha=1/2$. A family with one test then counts half as much as a family with two or more.

Baselines. All six baselines, like Ruleusion, see only the target's replies. SelfCheckGPT-NLI, the variant its authors recommend, and Semantic Entropy use $N=12$ samples, where their curves have flattened (Figure 3). Semantic Volume (Li et al., 2026) reuses Semantic Entropy's samples (`gte-Qwen2-1.5B-instruct`, PCA dimension 10, maximum length 64). MetaQA-MR (metamorphic relations) uses its published setup: three batched and three separate mutations of each relation. SAC^3-Q, the question-only variant, uses its published prompts and protocol with $k=10$ paraphrases (Zhang et al., 2023), and scores 0 when no paraphrase passes its equivalence check. The LLM judge is one greedy call to a separate GPT-4o-mini that sees q and a_0 and says whether a_0 is correct (NO counts as 1). The ablations keep the head in equation 3 and drop part of the evidence.

Inference and accounting. The original answer is generated once and shared by every method. Ruleusion and SAC^3 decode greedily, with completion caps of 8 tokens for a yes/no answer, 32 for a short fact, and 80 otherwise; SelfCheckGPT and Semantic Entropy sample at temperature 1.0. MetaQA spends eight calls writing mutations and twelve verifying them, and SAC^3 26.9 calls on average (Table 6). NLI, parsing, rule selection, and averaging run locally, and Ruleusion and SelfCheckGPT-NLI share one DeBERTa-v3-large MNLI checkpoint. Costs count the prompt and completion tokens of every call beyond the shared answer. Each method is scored on the answers for which it returned a score.

Labels, metrics, and tests. A label starts from a normalized match to the benchmark answer or to one of its aliases. If that match does not decide the row, we read the question, the gold aliases, and the original answer, and we do not look at any detector score. The positive class is the incorrect answer. A detector's threshold depends on the use, so we report the area under the receiver operating characteristic curve (AUROC), the rate at which a detector ranks a wrong answer above a right one. We compute it in each model–dataset cell and average models, then datasets, with equal weight, so that no dataset counts more for its size. All methods share the 25 cells, so significance uses a two-sided paired Wilcoxon signed-rank test on the per-cell differences, with Holm correction to control the family-wise error rate over the six comparisons; confidence intervals use a bootstrap over questions (Appendix F).

5 RESULTS

Overall comparison. Ruleusion has the highest five-dataset average AUROC, 0.744 (Table 2), and ranks first on FreshQA, Mintaka, and StrategyQA; SAC^3 leads on NQ and MetaQA on TruthfulQA. The area under the precision–recall curve (AUPRC) for the hallucination class gives the same leader on every dataset and on average (Appendix E). Over the 25 cells its mean paired lead is 2.2 points over SAC^3, 3.7 over MetaQA, 7.3 over SelfCheckGPT, 9.0 over Semantic Entropy, 9.2 over

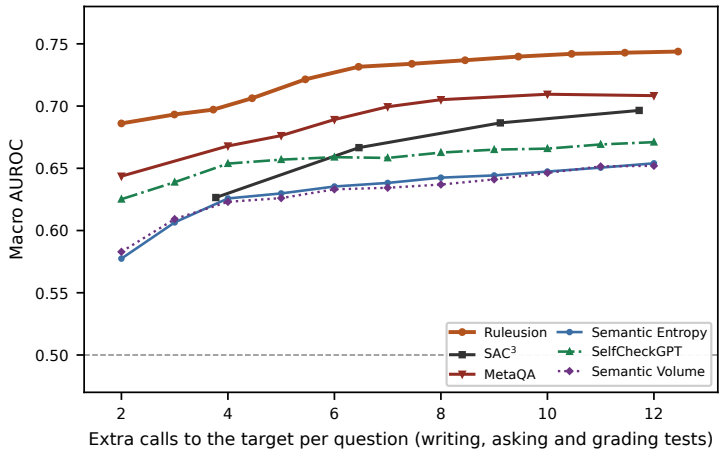


Figure 3: AUROC against extra calls to the target per question, averaged over 25 cells, counting every call that writes, asks, or grades a test; all curves are shown up to 13 calls. Ruleusion adds P1, the other probes, wobble, and then its first K rewrites; the sampling detectors use their first N samples, MetaQA its first m mutations, and SAC^3 its first k paraphrases. At their full budgets, SAC^3 (26.9 calls) and MetaQA (20 calls) reach 0.722 and 0.707 (Table 2). The dashed line is chance.

Table 3: Ruleusion against the two input-changing detectors, each at the smallest budget that gives it at least as many calls. Δ and its question-bootstrap 95% CI in AUROC points; p from a paired Wilcoxon test over 25 cells.

Ruleusion	Baseline (calls)	Ruleusion	Baseline	Δ	95% CI	Wins	p
K=2, 6.5 calls	SAC^3, k=3 (9.1)	.732	.686	+4.5	[+2.8, +6.2]	20/25	4×10^{-4}
	MetaQA, m=5 (7.0)	.732	.699	+3.2	[+1.2, +5.0]	19/25	0.024
Full, 12.5 calls	SAC^3, k=5 (14.3)	.744	.702	+4.2	[+2.5, +5.9]	21/25	5×10^{-4}
	MetaQA, m=9 (14.0)	.744	.708	+3.6	[+1.6, +5.2]	18/25	0.019

Semantic Volume, and 11.7 over the LLM judge (Table 5); the gains over the last five are significant after Holm correction. The gains over the sampling detectors come from grading each reply against an expected answer a rule fixed in advance. The largest margin by question type is on Mintaka's comparative questions, which the CONVERSE and ALT-DECOMP rules rewrite: 0.814, 8.4 points above the next detector (Appendix D).

Query and token cost. Ruleusion spends 12.5 calls and 870 tokens per question, SAC^3 26.9 calls and 2,934 tokens, and MetaQA 20.0 calls and 2,822 tokens (Table 6). **Ruleusion thus uses 54% fewer calls and 70% fewer tokens than SAC^3**, and 37% and 69% fewer than MetaQA. The difference lies in writing and grading: SAC^3 spends 18.9 calls and 2,406 tokens on them and MetaQA 8.0 calls and 1,691 tokens. Hand-written rules and local grading are what remove those writing and grading calls. With a call count close to resampling, each remaining call asks a new question.

Accuracy at matched budgets. The Ruleusion curve lies above every other curve at every budget (Figure 3); at 12.5 calls it scores higher than any other detector with up to 27 calls. Given at least as many calls as Ruleusion (Table 3), SAC^3 trails by 4.5 points at 6.5 calls and by 4.2 at 12.5, and MetaQA by 3.2 and 3.6. With K=2 rewrites plus the probes, 6.5 calls, Ruleusion already matches SAC^3 at its full 26.9 calls (0.732 against 0.722) and is above MetaQA at 20 calls and every sampling detector at twelve samples. The probes and wobble alone, 4.5 calls, reach 0.706, level with MetaQA at 20 calls. The curve levels off by K=6 (0.742, against 0.744 at K=8). The probes and the first rewrites already hold most of the score, so a larger set of checks adds little.

Table 4: Ablation of the two sources of tests. Mean AUROC over five models. Wobble is read from the probe replies and stays with the probes. With neither source there is no evidence, and AUROC is 0.5.

	FreshQA	NQ	Mintaka	StrategyQA	TruthfulQA	Avg.
Ruleusion (rewrites and probes)	.783	.760	.800	.690	.685	.744
Rewrites only	.799	.661	.719	.640	.624	.689
Probes only	.738	.722	.738	.665	.670	.706

What do the rewrites and the probes contribute? Table 4 keeps one of the two sources of tests of §3 at a time. The rewrites and the probes are complementary. Alone they reach 0.689 and 0.706; their scores correlate only weakly across the questions of a cell (mean Spearman $\rho = 0.32$); and each raises the other, by 5.5 points for probes on top of rewrites (20 of 25 cells) and 3.8 points for rewrites on top of probes (23 of 25). The two sources matter on different datasets. On FreshQA, whose facts change over time, the rewrites supply the evidence: alone they reach 0.799. On StrategyQA and TruthfulQA the probes supply it: without them the score falls by 0.050 and 0.061. The rewrites test whether answers to related questions fit the original answer, and the probes test the model's commitment to that answer; combined, they are more accurate than either alone on four of the five datasets. Removing the LOGIC layer, the rewrites that state a relation, lowers the score in 22 of 25 cells and on every dataset, from 0.744 to 0.734; on the questions where a LOGIC rule applies, it falls from 0.735 to 0.708 (23 of 25 cells). What the LOGIC layer adds is a stated relation, which tells the grader what reply would fit a right answer.

Where the evidence comes from. On 51.6% of answers (3,091 of 5,995), SelfCheckGPT's twelve samples normalize to one answer (Table 9). There SelfCheckGPT falls from 0.662 to 0.555 AUROC, Semantic Entropy from 0.641 to 0.532, and SAC^3 from 0.705 to 0.658, while Ruleusion keeps 0.693, against 0.694 on the other answers. **It is the only detector that keeps its accuracy there**, and it is 3.5 points above SAC^3 (95% CI $[+0.6, +6.5]$) and 13.8 and 16.1 points above SelfCheckGPT and Semantic Entropy; MetaQA falls slightly, from 0.686 to 0.674. The same holds when Semantic Entropy's own samples define collapse (Table 9). Collapse does not mean correct: 22.6% of collapsed answers are wrong. LOGIC rewrites and probes score each reply against an expected reply fixed before the query, so they still work when resampling has no disagreement left to read. Changing only the wording is not enough. With its WRAPPER rewrites alone, Ruleusion behaves like SelfCheckGPT: 0.661 on the other answers and 0.552 on collapsed ones. Its LOGIC rewrites and probes together keep 0.693 on collapsed answers, ahead of the WRAPPER rewrites within every dataset except FreshQA.

6 LIMITATIONS

Ruleusion flags an answer when a reply breaks a stated relation. When the replies stay consistent under those relations, as on TruthfulQA's misconception questions, check the answer against an outside source. For facts that change over time, as in FreshQA, use a source that carries a date. The uniqueness probe applies when the benchmark has one answer.

7 CONCLUSION

Ruleusion detects hallucinations without a reference answer by asking questions whose expected replies follow from a stated logical relation; no LLM writes or grades the tests. Across five models and five datasets it has the highest average AUROC of seven detectors, 0.744, and the best accuracy at every call budget, with 54% fewer calls than SAC^3 and 37% fewer than MetaQA. Where all twelve samples coincide, it keeps 0.69 AUROC, the only detector that keeps its accuracy there.

ETHICS STATEMENT

False positives and missed errors can affect decisions that depend on the score. Where a decision has consequences, the score should be read together with outside evidence and a human review.

Unresolved matches are read by the authors from the question, the gold aliases, and the original answer, without detector scores. We log the flagged rows and the reasons given for them, so that the labels can be redone independently.

Reproducibility statement

Rule templates, probe templates, prompts, the scoring head (equation 3), and the per-question scores of every method are included as supplementary material. Appendix F records the analysis seed and the FreshQA label rule.

AI use statement

We used generative AI tools for (i) language editing and conversion of the manuscript to the conference format, and (ii) retrieval and discovery of related work, where every suggested reference was located and read by the authors before being cited. We did not use generative AI tools to generate synthetic data, develop the theoretical framework, formulate or prove mathematical claims, propose hypotheses, design experiments, implement methods, clean or reformat datasets, or interpret results. Separately from authoring assistance, the study itself evaluates LLM-generated answers (including responses to deterministic variants and probes) and includes an LLM-judge baseline. Unresolved benchmark matches are read by the authors from the question, the gold aliases, and the original answer, without detector scores, as described in §4. Equations, figures, and reported numbers were checked against the released scores. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Tauman Kalai. Do language models know when they're hallucinating references? In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 912–928, 2024.

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

Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys*, 51(1):1–27, 2018.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Unified hallucination detection for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3235–3252, 2024b.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. FacTool: Factuality detection in generative AI—a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*, 2023.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1419–1436, 2024.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 12621–12640, 2023.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, 2024.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

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

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Bairu Hou, Yang Zhang, Jacob Andreas, and Shiyu Chang. A probabilistic framework for LLM hallucination detection via belief tree propagation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3076–3099, 2025. doi: 10.18653/v1/2025.naacl-long.158.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Ningke Li, Yuekang Li, Yi Liu, Ling Shi, Kailong Wang, and Haoyu Wang. Drowzee: Metamorphic testing for fact-conflicting hallucination detection in large language models. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA2), 2024. doi: 10.1145/3689776.

Xiaomin Li, Zhou Yu, Ziji Zhang, Yingying Zhuang, Swair Shah, Narayanan Sadagopan, and Anurag Beniwal. Semantic volume: Quantifying and detecting both external and internal uncertainty in LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31751–31759, 2026.

Buyun Liang, Liangzu Peng, Jinqi Luo, Darshan Thaker, Kwan Ho Ryan Chan, and René Vidal. SECA: Semantically equivalent and coherent attacks for eliciting LLM hallucinations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3214–3252, 2022.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9004–9017, 2023.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 12076–12100, 2023.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4902–4912, 2020.

Sergio Segura, Gordon Fraser, Ana B. Sanchez, and Antonio Ruiz-Cortés. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, 2016.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pp. 1604–1619, 2022.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. LLM-Check: Investigating detection of hallucinations in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Neeraj Varshney, Satyam Raj, Venkatesh Mishra, Agneet Chatterjee, Amir Saeidi, Ritika Sarkar, and Chitta Baral. Investigating and addressing hallucinations of LLMs in tasks involving negation. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pp. 580–598, 2025. doi: 10.18653/v1/2025.trustnlp-main.37.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, 2024.

Weibin Wu, Yuhang Cao, Ning Yi, Rongyi Ou, and Zibin Zheng. Detecting and reducing the factual hallucinations of large language models with metamorphic testing. *Proceedings of the ACM on Software Engineering*, 2(FSE):1432–1453, 2025. doi: 10.1145/3715784.

Borui Yang, Md Afif Al Mamun, Jie M. Zhang, and Gias Uddin. Hallucination detection in large language models with metamorphic relations. *Proceedings of the ACM on Software Engineering*, 2(FSE):425–445, 2025.

Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. InterrogateLLM: Zero-resource hallucination detection in LLM-generated answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9333–9347, 2024. doi: 10.18653/v1/2024.acl-long.506.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. SAC3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15445–15458, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

A SIGNIFICANCE TESTS

Table 5: Paired comparison of Ruleusion against each baseline over the 25 per-cell AUROC differences (Δ = Ruleusion $-$ baseline). Two-sided Wilcoxon signed-rank test, Holm correction over the six tests; bootstrap resamples questions jointly within each cell (1,000 draws, seed 0). Ruleusion macro AUROC 0.744, 95% CI $[0.729, 0.758]$. On the five dataset means, Ruleusion is ahead of SAC^3 and MetaQA on four datasets and of every other baseline on all five. Table 3 compares Ruleusion with SAC^3 and MetaQA at matched call budgets.

Baseline	Mean Δ	Wins	Wilcoxon p	Holm p	Bootstrap 95% CI of Δ
SAC^3	$+0.022$	16/25	0.052	0.052	$[+0.006, +0.039]$
MetaQA	$+0.037$	18/25	0.020	0.041	$[+0.020, +0.054]$
SelfCheckGPT	$+0.073$	22/25	5×10^{-6}	2×10^{-5}	$[+0.057, +0.091]$
Semantic Entropy	$+0.090$	23/25	4×10^{-7}	2×10^{-6}	$[+0.075, +0.106]$
Semantic Volume	$+0.092$	23/25	3×10^{-7}	1×10^{-6}	$[+0.074, +0.109]$
LLM judge	$+0.117$	24/25	1×10^{-7}	7×10^{-7}	$[+0.101, +0.135]$

B COST BREAKDOWN

Table 6: Extra calls and tokens per question, averaged over 25 cells. A call writes a test, asks one, or grades a reply; the shared original answer is excluded for every method, and no method is charged for local computation. Ruleusion spends 8.0 calls on rewrites (539 tokens) and 4.5 calls on probes (331 tokens), and the model writes 45 output tokens per question. Semantic Volume is computed from Semantic Entropy's samples. †One call to a separate judge model (GPT-4o-mini).

	Writing		Asking		Grading		Total		Tokens
Method	calls	tokens	calls	tokens	calls	tokens	calls	tokens	/ call
Ruleusion	0	0	12.5	870	0	0	12.5	870	69
SAC^3	10.9	1,285	8.0	528	8.0	1,121	26.9	2,934	109
MetaQA	8.0	1,691	12.0	1,131	0	0	20.0	2,822	141
SelfCheckGPT	0	0	12.0	764	0	0	12.0	764	64
Semantic Entropy	0	0	12.0	764	0	0	12.0	764	64
LLM judge†	0	0	0	0	1.0	36	1.0	36	36

C PER-CELL RESULTS

Table 7: AUROC per model–dataset cell for all seven methods. SE = Semantic Entropy, SV = Semantic Volume.

Model	Dataset	n	Ruleusion	SAC^3	MetaQA	SelfCheck	SE	SV	Judge
DeepSeek	FreshQA	99	.846	.837	.773	.852	.859	.744	.686
GPT-4o-mini	FreshQA	99	.775	.711	.725	.723	.764	.716	.627
Qwen3-Max	FreshQA	99	.742	.749	.783	.696	.640	.561	.654
Llama-3.3-70B	FreshQA	99	.782	.720	.744	.730	.737	.702	.726
Mistral-Large	FreshQA	99	.769	.720	.733	.680	.644	.635	.625
DeepSeek	NQ	100	.885	.867	.721	.768	.785	.780	.762
GPT-4o-mini	NQ	100	.758	.743	.713	.693	.711	.765	.592
Qwen3-Max	NQ	100	.652	.745	.640	.505	.576	.548	.538
Llama-3.3-70B	NQ	100	.763	.776	.644	.737	.696	.736	.615
Mistral-Large	NQ	100	.744	.730	.632	.614	.639	.644	.619
DeepSeek	Mintaka	300	.813	.821	.739	.768	.734	.710	.646
GPT-4o-mini	Mintaka	300	.771	.769	.746	.787	.777	.753	.630
Qwen3-Max	Mintaka	300	.805	.809	.694	.714	.676	.707	.651
Llama-3.3-70B	Mintaka	300	.788	.810	.702	.770	.723	.733	.645
Mistral-Large	Mintaka	300	.824	.749	.726	.674	.639	.683	.630
DeepSeek	StrategyQA	500	.759	.780	.636	.594	.607	.655	.625
GPT-4o-mini	StrategyQA	500	.692	.655	.613	.622	.572	.533	.529
Qwen3-Max	StrategyQA	500	.660	.679	.638	.614	.548	.604	.560
Llama-3.3-70B	StrategyQA	500	.753	.672	.709	.669	.593	.623	.616
Mistral-Large	StrategyQA	500	.587	.575	.675	.644	.524	.603	.595
DeepSeek	TruthfulQA	200	.666	.686	.766	.586	.583	.596	.623
GPT-4o-mini	TruthfulQA	200	.670	.554	.679	.566	.580	.530	.630
Qwen3-Max	TruthfulQA	200	.736	.658	.741	.597	.623	.554	.596
Llama-3.3-70B	TruthfulQA	200	.717	.626	.740	.606	.528	.574	.657
Mistral-Large	TruthfulQA	200	.638	.607	.769	.567	.578	.615	.584

D QUESTION-TYPE SLICES AND COLLAPSE DIAGNOSTIC

Table 8 splits Mintaka by complexity type. Ruleusion ranks first on four of the nine types, more than any other detector, and its largest margin is on comparative questions: 0.814, 8.4 points above the next detector (SAC^3, 0.730). The CONVERSE and ALT-DECOMP rules in Table 1 rewrite exactly

these questions. SAC^3 leads on count, difference, and multihop questions. SelfCheckGPT leads on superlative questions, and MetaQA on yes/no questions, where 33 questions per model and a hallucination rate of 0.109 leave about four wrong answers per model.

Table 8: Mintaka AUROC by complexity type, computed within each model and averaged over the five models; n = questions per type per model.

Type	n	Halluc. rate	Ruleusion	SAC^3	MetaQA	SelfCheck	SE	SV	Judge
comparative	34	.194	**.814**	.730	.714	.549	.571	.466	.624
count	34	.388	.771	**.777**	.622	.727	.674	.711	.660
difference	34	.465	.656	**.726**	.645	.725	.636	.620	.579
generic	33	.255	**.812**	.787	.672	.503	.650	.642	.615
intersection	33	.261	**.849**	.818	.758	.824	.758	.832	.728
multihop	33	.521	.815	**.861**	.731	.765	.723	.692	.549
ordinal	33	.327	**.828**	.815	.746	.729	.730	.685	.746
superlative	33	.364	.694	.657	.696	**.736**	.713	.716	.641
yes/no	33	.109	.845	.754	**.866**	.857	.661	.582	.588

Table 9: Collapse diagnostic, pooled over answers. k_{unique} is the number of distinct normalized answers among SelfCheckGPT's twelve samples. The hallucination rate rises with k_{unique} (0.23 at k=1, 0.37 at k=2, 0.58 at k=6, 0.71 at k=12). When Semantic Entropy's own samples define collapse (3,103 answers; Semantic Entropy and Semantic Volume, which read these samples, are left out), the AUROCs are 0.692 for Ruleusion, 0.657 for SAC^3, 0.680 for MetaQA, 0.583 for SelfCheckGPT, and 0.592 for the judge.

Subset	n	Halluc. rate	Ruleusion	SAC^3	MetaQA	SelfCheck	SE	SV	Judge
Collapsed (k_{unique}=1)	3,091	.226	.693	.658	.674	.555	.532	.588	.590
Not collapsed (k_{unique}>1)	2,904	.465	.694	.705	.686	.662	.641	.613	.622

E AUPRC

Table 10: Area under the precision–recall curve for the hallucination class, computed per cell and averaged as in Table 2. The last row, the share of wrong answers, is the AUPRC of a random score.

Method	FreshQA	NQ	Mintaka	StrategyQA	TruthfulQA	Avg.
Ruleusion	**.775**	.689	**.647**	**.429**	.629	**.634**
SAC^3	.740	**.700**	.638	.417	.552	.609
MetaQA	.739	.593	.511	.379	**.687**	.582
SelfCheckGPT	.756	.618	.609	.371	.546	.580
Semantic Entropy	.717	.600	.550	.321	.532	.544
Semantic Volume	.678	.635	.570	.338	.524	.549
LLM judge	.633	.524	.429	.320	.531	.487
Hallucination rate	.507	.420	.321	.266	.444	.392

F Reproducibility notes

The question-level bootstrap in Appendix A and Table 3 uses seed 0 and 1,000 resamples, as does the interval for the collapse comparison in §5, which resamples questions and keeps each question's answers together. Variant and probe calls use greedy decoding. SelfCheckGPT and Semantic Entropy sample at temperature 1.0. The NLI encoder is `potsawee/deberta-v3-large-mnli`.

FreshQA is scored against the gold aliases of Vu et al. (2024). Alias matching left 232 of 495 questions unresolved. We reviewed those questions with the question, the gold aliases, and the original answer, and without detector scores. On the questions resolved by alias matching, mean AUROC is 0.861 for Ruleusion and 0.748 for MetaQA, with 48–57 questions per model.

G THRESHOLD SWEEP

MetaQA reports precision, recall, and F1 at a fixed threshold. Table 11 applies the same protocol to both methods on the pooled 5,995 answers (score $\geq \theta$ predicts hallucination). Ruleusion's best F1 on the grid is 0.600 at θ=0.20; MetaQA's is 0.573 at θ=0.10; predicting every answer as hallucinated gives 0.510.

Table 11: Precision, recall, and F1 for the hallucination class at fixed thresholds, pooled over 25 cells (hallucination rate 0.342). MetaQA's score is one minus its metamorphic-relation consistency.

	Ruleusion			MetaQA		
θ	P	R	F1	P	R	F1
0.10	.408	.920	.565	.460	.759	**.573**
0.20	.490	.773	**.600**	.485	.676	.565
0.30	.551	.529	.540	.507	.603	.551
0.40	.599	.332	.427	.529	.529	.529
0.50	.615	.205	.307	.563	.438	.493
0.60	.645	.107	.184	.635	.199	.303
0.70	.639	.049	.091	.664	.157	.254