

Hallucination-R1: Robustness-Oriented Paraphrase Generation for Factual Consistency

Wenhan Yu* Wenxin Wu* Hao Wang Lei Sha†

Beihang University

Beijing, 100191, China

yuwenhan@buaa.edu.cn wwx20030520@buaa.edu.cn

wanghao_ai@buaa.edu.cn shalei@buaa.edu.cn

Abstract

Factual hallucination is commonly defined by incorrect factual outputs. We study a paraphrase-induced hallucination setting, where a model answers a factual question correctly in its original form but generates an incorrect answer under a semantically equivalent paraphrase. Such inconsistencies expose latent factual instability under semantic invariance. However, general-purpose paraphrases are often insufficient as robustness-oriented supervision: near-copy paraphrases provide weak signals, while overly diverse paraphrases may break semantic equivalence. In this paper, we propose HALLUCINATION-R1, a robustness-oriented paraphrase generation framework that learns to produce semantically faithful yet robustness-challenging paraphrases for factual consistency. Through two-stage optimization, it first stabilizes meaning-preserving and diverse paraphrasing, then rewards paraphrases that reveal factual consistency degradation in downstream QA models. Experiments on SimpleQuestions, PopQA, and TruthfulQA show that HALLUCINATION-R1 achieves a strong consistency–diversity trade-off and exposes robustness failures across multiple model families and datasets. Further analyses indicate that these failures are not reducible to surface-level artifacts or semantic drift, but reveal non-trivial factual instability under meaning-preserving variation. A lightweight fine-tuning study also shows that HALLUCINATION-R1-generated data improves robust accuracy under paraphrase variations, suggesting its utility for robustness-oriented training. Our code and models are publicly available at <https://github.com/yuwenhan07/Hallucination-R1>.

1 Introduction

Large language models (LLMs) have shown strong question-answering capabilities, yet their factual

behavior can remain unstable under semantics-preserving paraphrases (Brown et al., 2020; Singh et al., 2025). A model may answer an original factual question correctly, but produce an incorrect response when the same query is expressed in a different surface form (Ribeiro et al., 2018; Gan and Ng, 2019). Since the factual intent remains unchanged, such failures suggest that hallucination is not merely caused by knowledge absence, but can also reflect unstable factual behavior, where the model’s internal preference over known facts drifts under meaning-preserving linguistic variation (Lin et al., 2022; Huang et al., 2025).

This motivates a shift from treating paraphrases merely as diagnostic probes (Ribeiro et al., 2020) to learning paraphrase distributions that can provide useful supervision for improving factual robustness. For robustness-oriented fine-tuning, such data should preserve the original factual intent while introducing meaningful linguistic variation, enabling models to learn stable factual behavior across different surface forms. This is difficult because near-copy paraphrases provide limited robustness signals, whereas overly aggressive paraphrases may change the underlying factual query (Feng et al., 2021).

In this paper, we propose HALLUCINATION-R1, a robustness-oriented paraphrase generation framework for factual consistency. Instead of constructing arbitrary adversarial prompts, HALLUCINATION-R1 learns to generate semantically consistent yet robustness-challenging paraphrases for factual question answering. The framework follows a two-stage optimization pipeline: it first learns to generate semantically valid and diverse paraphrases, and then further optimizes them to expose factual consistency degradation in downstream QA models under semantic invariance.

Experiments on SimpleQuestions, PopQA, and TruthfulQA show that HALLUCINATION-R1 generates semantically consistent and diverse para-

*These authors contributed equally to this work.

†Corresponding author.

phrases that expose factual robustness failures across multiple QA models, including settings beyond the training dataset and feedback model. Further analyses suggest that these failures are non-trivial, and a lightweight fine-tuning study demonstrates the utility of the generated data for improving robust accuracy under paraphrase variations.

Our contributions are summarized as follows:

- We formulate robustness-oriented paraphrase generation for factual consistency, where paraphrases must preserve semantics while exposing factual instability under surface-form variation.
- We propose HALLUCINATION-R1, a two-stage GRPO-based framework that combines semantic consistency, group-relative diversity, and factual inconsistency exposure signals.
- We show that HALLUCINATION-R1 exposes factual robustness failures across multiple QA models and datasets, and that these failures reflect non-trivial factual instability rather than simple surface-level artifacts.
- We demonstrate the training utility of HALLUCINATION-R1-generated paraphrases, as they improve robust accuracy under semantics-preserving paraphrase variations.

2 Related Work

2.1 Factual Hallucination and Paraphrase Inconsistency

Factual hallucination broadly refers to cases where large language models generate outputs that are inconsistent with real-world facts or unverifiable, including fabricated events, incorrect dates, or non-existent references (Huang et al., 2025; Ji et al., 2023). Prior studies attribute such errors to factors including biased or incomplete training data, outdated knowledge, and overconfident decoding behaviors (Li et al., 2024; Ji et al., 2023). Recent work further shows that language models may produce highly misleading factual errors even when semantic content and surface-level fluency are largely preserved, posing substantial risks to high-reliability applications (Zhang et al., 2023; Lin et al., 2022).

In contrast to settings where factual errors primarily indicate missing, outdated, or inaccessible knowledge, this work focuses on cases where the model answers the original question correctly but fails under a semantically equivalent paraphrase.

Such failures suggest that the model may possess the relevant factual knowledge, yet fail to maintain a stable factual response under meaning-preserving linguistic variation (Manakul et al., 2023; Choi, 2025). Since directly determining whether a model truly "knows" a fact is inherently difficult, inconsistencies across semantically equivalent paraphrases provide a practical signal for characterizing latent factual instability (Choi, 2025).

2.2 Paraphrase Robustness Evaluation

Recent studies have shown that language models are highly sensitive to semantics-preserving reformulations (Gan and Ng, 2019; Choi, 2025). Even when the underlying factual intent remains unchanged, paraphrase variations can lead to substantial performance degradation or inconsistent predictions (Lunardi et al., 2025). Existing work mainly uses paraphrases as evaluation probes, measuring whether model predictions remain consistent across semantically equivalent question forms (Gan and Ng, 2019; Choi, 2025). These studies reveal the brittleness of current language models under paraphrase variation.

However, existing paraphrase-based evaluation and training studies typically construct paraphrases through predefined paraphrasing strategies, similarity filters, or diversity-oriented criteria. Although recent work on worst-case prompting and latent adversarial robustness further highlights the importance of challenging input variations (Cao et al., 2024; Fu and Barez, 2025), existing paraphrase construction largely focuses on semantic preservation and linguistic diversity, without explicitly targeting paraphrase patterns that expose factual robustness fragility. This leaves open the question of how to generate paraphrases that are not only semantically valid, but also effective for revealing factual instability and supporting robustness-oriented training.

2.3 Robustness-Oriented Paraphrase Generation

Existing paraphrase generation and text augmentation methods provide useful mechanisms for producing meaning-preserving or linguistically diverse paraphrases (Huang et al., 2023; Feng et al., 2021). Building on the gap above, we argue that factual robustness requires a more targeted paraphrase objective. A useful paraphrase should preserve the answer-bearing intent while exposing whether a QA model can maintain factual consistency across meaning-preserving variations.

We therefore formulate paraphrase generation as a robustness-oriented learning problem. HALLUCINATION-R1 learns to generate semantically constrained paraphrases that preserve factual intent while exposing factual consistency degradation in QA models, making them useful for both robustness evaluation and training.

3 Methodology

We aim to generate robustness-oriented paraphrases for factual question answering. A useful paraphrase should remain semantically equivalent to the original question, introduce diverse surface-form variations, and reveal whether a QA model can maintain factual consistency under meaning-preserving reformulations. HALLUCINATION-R1 therefore treats paraphrase generation as a constrained robustness-oriented learning problem.

As shown in Figure 1, we train a dedicated paraphrase generator through a two-stage optimization pipeline. The first stage stabilizes semantic-preserving paraphrase generation with controlled diversity, while the second stage further encourages paraphrases that expose factual inconsistency in downstream QA models.

3.1 Semantic-Preserving Challenging Paraphrase Generation

Given an original factual question x , we train a paraphrase generator π_θ to produce a paraphrase $y \sim \pi_\theta(\cdot | x)$. We decompose the optimization signal into three components: semantic consistency, paraphrase diversity, and factual inconsistency exposure.

Semantic consistency. Semantic consistency is enforced as a hard constraint in our framework. We use an LLM-as-judge M_{consis} to determine whether a paraphrase preserves the original meaning. In our reward design, semantic consistency serves as a hard gate: if a paraphrase is judged semantically inconsistent, its overall reward is set to zero regardless of other signals. We denote the corresponding consistency reward as R_{consis} .

Diversity. To prevent semantically consistent paraphrases from collapsing into near copies, we introduce a group-relative diversity reward R_{div} . For each question, we sample a group of candidate paraphrases and assign rank-based rewards according to the lexical and semantic dissimilarity between each semantically consistent candidate and the others in the same group. This group-relative scoring

encourages exploration by rewarding candidates that are more diverse than their peers, rather than those that merely pass a fixed diversity threshold. Details are provided in Appendix A.1.

Factual inconsistency exposure. We further encourage the generator to produce paraphrases that reveal factual instability in downstream QA models. For each paraphrase, the QA model first generates an answer z , and an LLM-as-judge M_{halluc} evaluates whether z is factually inconsistent with the ground-truth answer. The judge outputs a factual inconsistency score $R_{\text{halluc}} \in \{0, 1, 2\}$, where higher values indicate stronger factual inconsistency. Details are provided in Appendix A.2.

Sequence-level optimization. Our objective naturally fits GRPO, since diversity is inherently a group-level property: it can only be measured across multiple paraphrases rather than from a single output. GRPO samples a group of candidate paraphrases and optimizes them with group-relative rewards, aligning well with our diversity reward while avoiding an explicit value function. The objective is:

$$\max_{\theta} \mathbb{E}_{y \sim \pi_\theta(\cdot | x)} [R(x, y, z)] - \beta D_{\text{KL}}(\pi_\theta(y | x) \| \pi_{\text{ref}}(y | x)), \quad (1)$$

where $z \sim \pi_{\text{QA}}(\cdot | y)$ denotes the answer generated by the QA model, π_{ref} is the reference policy, and β controls the strength of KL regularization.

3.2 Two-Stage Training Pipeline

Unless otherwise specified, HALLUCINATION-R1 is initialized from Qwen2.5-3B, and downstream QA feedback during training is obtained from Qwen2.5-3B as well. We use DeepSeek-V3 as the LLM judge for both semantic consistency verification and factual inconsistency scoring, corresponding to M_{consis} and M_{halluc} , respectively.

Semantic Stabilization Stage. Before optimizing factual inconsistency exposure, we first stabilize semantic-preserving paraphrase generation, since the semantic-consistency hard gate would otherwise make downstream QA feedback highly sparse. The first-stage reward is defined as:

$$R_{\text{stab}} = R_{\text{consis}} + \lambda_{\text{div}} R_{\text{div}}, \quad (2)$$

where λ_{div} controls the strength of diversity exploration.

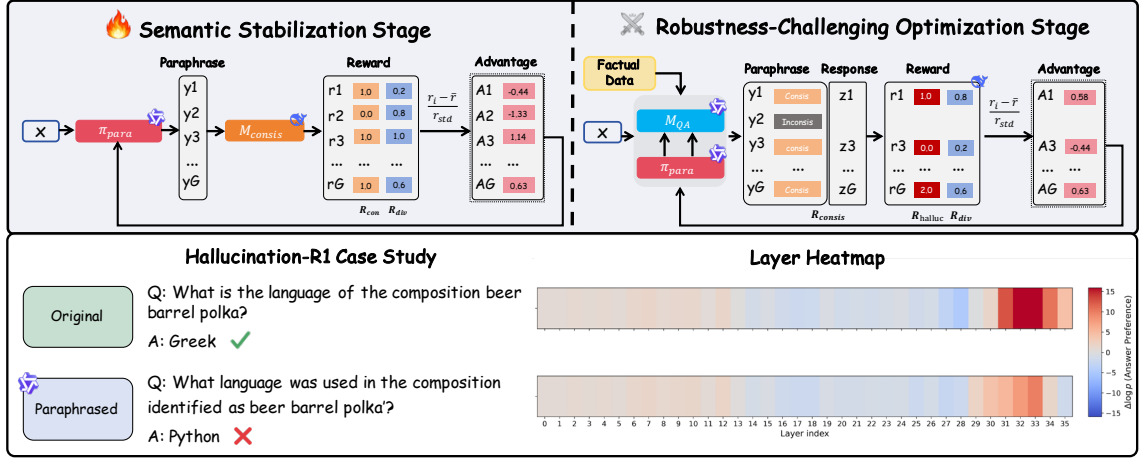


Figure 1: **Overview of the HALLUCINATION-R1 framework.** The upper panel illustrates the two-stage training pipeline: the **semantic stabilization stage** learns semantically preserved paraphrases with controlled diversity exploration, while the **robustness-challenging optimization stage** optimizes paraphrases to expose factual inconsistency in a QA model under semantic invariance. The lower panel shows a representative **case study**, where a semantically equivalent paraphrase shifts the QA model from a correct answer to a factual hallucination. The heatmap visualizes layer-wise log-probability preference changes, with red indicating stronger preference for the correct answer token and blue for the incorrect one.

Robustness-Challenging Optimization Stage.

After semantic stabilization, we further optimize the generator to expose factual inconsistency in downstream QA models. Given a generated paraphrase y whose original question x is correctly answered by the QA model, the QA model produces an answer z , and the LLM-as-judge M_{halluc} evaluates whether z is factually inconsistent with the ground-truth answer. The training reward is:

$$R_{\text{train}} = \begin{cases} R_{\text{stab}} + R_{\text{halluc}}, & R_{\text{halluc}} \neq 0, \\ 0, & R_{\text{halluc}} = 0, \end{cases} \quad (3)$$

Semantic consistency remains a hard constraint throughout training.

4 Consistency–Diversity Trade-off Analysis

We analyze the consistency–diversity trade-off across different paraphrasing models, including prompted LLM paraphrasing baselines. The consistency and diversity metrics follow the same definitions used in training rewards.

Table 1 shows clear consistency–diversity differences across paraphrasing models. Some models achieve strong semantic consistency but produce limited expression variation, while others generate more diverse paraphrases at the cost of semantic stability. This imbalance is evident across model families: the Qwen3 series tends to produce conservative or near-copy paraphrases to maintain ex-

tremely high consistency, whereas Qwen2.5-3B is more flexible but often sacrifices semantic stability when increasing diversity. These patterns highlight the need for targeted optimization rather than relying on general-purpose paraphrasing.

In contrast, large models such as DeepSeek-V3 and Qwen2.5-32B already achieve strong consistency–diversity performance under prompting alone, as reflected by high Consis. $\times$ Mean Div. scores. HALLUCINATION-R1 achieves competitive results using only a 3B-scale model and produces substantially larger proportions of high-diversity paraphrases. This suggests that robustness-oriented paraphrase optimization can reshape the generation distribution of a smaller model toward semantically valid yet more varied paraphrases.

The comparison between the Stage-1-only model and the full HALLUCINATION-R1 further illustrates the role of the two-stage design. The Semantic Stabilization Stage improves the overall consistency–diversity balance mainly by increasing semantic consistency, despite a modest drop in diversity. After this stabilization, the robustness-challenging optimization stage further enhances diversity exploration under semantic-preserving constraints. This pattern indicates that semantic stabilization provides a reliable basis for later exploration, rather than simply encouraging arbitrary surface variation.

Taken together, these results consistently sug-

Paraphrase Model	Consis.	Mean Div.	Ratio (Div. > 0.9)	Consis. $\times$ Mean Div.
Hallucination-R1-3B	87.46%	0.930696	80.41%	81.40%
DeepSeek-V3	96.41%	0.880482	55.19%	84.89%
Qwen2.5-3B	64.68%	0.900111	74.41%	58.22%
Qwen2.5-32B	96.03%	0.892318	74.41%	85.69%
Qwen3-4B	98.28%	0.399141	3.16%	39.23%
Qwen3-8B	98.71%	0.47959	5.11%	47.34%
Hallucination-R1 (Stage 1 Only)	80.37%	0.864836	58.56%	69.51%

Table 1: Consistency–diversity trade-off across different paraphrase models on the TruthfulQA dataset. We report semantic consistency (Consis.), mean diversity (Mean Div.), consistency $\times$ diversity, and the ratio of high-diversity paraphrases (Div. > 0.9). Semantic consistency follows the definition used in the training reward.

gest that HALLUCINATION-R1 yields a favorable consistency–diversity trade-off for robustness-oriented paraphrase generation. While consistency and diversity capture basic paraphrase quality, the next section further evaluates whether the learned distribution is also effective at exposing downstream factual robustness failures.

5 Robustness Evaluation

We evaluate whether semantically equivalent yet challenging paraphrases generated by HALLUCINATION-R1 can expose factual consistency degradation across different QA models, architectures, and benchmark datasets.

We train HALLUCINATION-R1 on a filtered subset of SimpleQuestions with reliable factual correctness and evaluate it on SimpleQuestions, PopQA, and TruthfulQA. We consider two model scales, 1.5B and 3B, based on Qwen2.5-1.5B and Qwen2.5-3B, respectively, and train them using the two-stage pipeline described earlier. Full experimental details are provided in Appendix C.

5.1 Evaluation Protocol

We evaluate robustness only on original questions that the QA model answers correctly. For each question, we generate K paraphrases using π_{para} , verify semantic equivalence with M_{consis} , and evaluate the QA model’s answer against the ground-truth answer using M_{judge} . A robustness failure is counted only when the paraphrase is semantically equivalent to the original question and causes the QA model to produce an incorrect answer. We set $K = 1$ for ER@1 and $K = 5$ for WER@5. The detailed procedure is provided in Appendix C.4.

We report two complementary robustness metrics:

- **Error Rate@1 (ER@1)**: the proportion of originally correct samples for which a single semantically equivalent paraphrase causes an incorrect answer.
- **Worst-Case Error Rate@5 (WER@5)**: the proportion of originally correct samples for which at least one of five generated paraphrases is both semantically equivalent and answered incorrectly.

5.2 Robustness Evaluation Results

Table 2 reports the robustness impact of paraphrases generated by HALLUCINATION-R1 under the strict constraint of semantic equivalence. Although the paraphrase generator is trained only on SimpleQuestions (Bordes et al., 2015) with Qwen2.5-3B as the QA model, it consistently exposes factual consistency degradation across target models of different scales and architectures, as well as on distributionally distinct datasets such as PopQA (Mallen et al., 2023) and TruthfulQA (Lin et al., 2022).

Across evaluation settings, semantically equivalent paraphrases substantially reduce factual robustness under surface-form variation. This effect is pronounced in Worst-Case Error Rate@5, where many originally correct samples become incorrect under at least one of multiple meaning-preserving paraphrases. Moreover, the degradation remains observable even for larger and stronger models: while higher-capacity models generally show lower single-paraphrase error rates, they still exhibit non-negligible worst-case robustness failures. These results suggest that factual consistency under paraphrase variation remains a systematic challenge, and that HALLUCINATION-R1 provides effective robustness probes for evaluating factual stability under semantic-preserving transformations.

QA Model	SimpleQuestions			PopQA			TruthfulQA		
	True (Acc)	ER@1	WER@5	True (Acc)	ER@1	WER@5	True (Acc)	ER@1	WER@5
Hallucination-R1-1.5B									
Qwen2.5-3B	2106(21.25%)	30.44%	71.37%	1702(11.93%)	30.14%	77.38%	246(31.14%)	32.11%	83.33%
Qwen2.5-7B	2288(23.09%)	29.98%	66.35%	2077(14.56%)	29.47%	73.57%	323(40.89%)	30.96%	73.07%
Qwen2.5-32B	2584(26.07%)	23.30%	56.93%	2368(16.60%)	25.13%	66.17%	374(47.34%)	24.60%	68.18%
Qwen3-4B	2126(21.45%)	28.46%	68.30%	1194(8.37%)	36.60%	86.43%	233(29.49%)	36.91%	81.55%
LLaMA-3.1-8B	2489(25.12%)	23.66%	56.77%	3082(21.60%)	22.97%	62.78%	247(31.27%)	29.15%	67.61%
DeepSeek-V3	3413(34.44%)	32.93%	46.35%	4661(32.67%)	36.84%	48.27%	391(49.49%)	22.51%	39.13%
DeepSeek-R1	4258(42.97%)	24.10%	37.48%	4560(31.96%)	43.75%	57.08%	466(65.73%)	20.82%	33.48%
GPT-5	5437(54.86%)	9.82%	25.31%	9912(69.48%)	7.31%	13.33%	645(82.78%)	4.50%	11.94%
Hallucination-R1-3B									
Qwen2.5-3B	2106(21.25%)	40.12%	86.70%	1702(11.93%)	37.60%	82.20%	246(31.14%)	47.97%	86.99%
Qwen2.5-7B	2288(23.09%)	36.80%	73.73%	2077(14.56%)	37.31%	81.51%	323(40.89%)	45.51%	85.76%
Qwen2.5-32B	2584(26.07%)	34.64%	71.17%	2368(16.60%)	38.05%	81.88%	374(47.34%)	48.40%	86.90%
Qwen3-4B	2126(21.45%)	38.90%	75.68%	1194(8.37%)	44.64%	90.28%	233(29.49%)	49.79%	87.12%
LLaMA-3.1-8B	2489(25.12%)	34.83%	71.11%	3082(21.60%)	36.53%	80.89%	247(31.27%)	55.06%	87.04%
DeepSeek-V3	3413(34.44%)	39.76%	51.36%	4661(32.67%)	46.34%	60.78%	391(49.49%)	41.69%	54.22%
DeepSeek-R1	4258(42.97%)	38.14%	50.28%	4560(31.96%)	49.76%	65.37%	466(65.73%)	41.85%	55.15%
GPT-5	5437(54.86%)	9.91%	26.04%	9912(69.48%)	8.34%	17.62%	645(82.78%)	6.20%	14.57%

Table 2: Robustness evaluation results of HALLUCINATION-R1 across different QA models and benchmark datasets. True (Acc) reports the number and percentage of samples that are correctly answered by the QA model before paraphrasing. ER@1 and WER@5 denote Error Rate@1 and Worst-Case Error Rate@5, respectively. A robustness failure is counted only when the rewritten question is judged to be semantically equivalent to the original input while causing the QA model to generate an incorrect answer. All robustness metrics are computed over the subset of samples that are originally answered correctly by the QA model. Results are reported for two HALLUCINATION-R1 scales: 1.5B and 3B parameters.

5.3 Ablation Study

To analyze the contribution of different training stages in HALLUCINATION-R1, we conduct ablation experiments on the TruthfulQA benchmark, with results summarized in Table 3. We compare the full HALLUCINATION-R1 model with two ablated variants: a model trained only with the Semantic Stabilization Stage, denoted as HALLUCINATION-R1 (Stage 1 Only), and a base paraphrasing model without reinforcement learning optimization (Qwen2.5-3B).

The results show that the Semantic Stabilization Stage provides an important foundation for generating semantically consistent paraphrases, while the subsequent Robustness-Challenging Optimization Stage substantially improves the effectiveness of paraphrases in exposing factual inconsistency under semantic-preserving transformations.

In addition, we compare the trained paraphrase generator with larger but untrained paraphrasing models such as Qwen2.5-7B. Despite its smaller scale, HALLUCINATION-R1-3B consistently produces more effective robustness-challenging paraphrases, indicating that the observed gains arise primarily from the proposed training strategy rather than model scale alone. Additional com-

parisons with larger paraphrase generators show similar trends, with detailed results reported in Appendix D.

These findings suggest that robustness-oriented paraphrase generation requires balancing semantic fidelity and robustness-challenging effectiveness.

6 Diagnostic Analysis of Robustness Failures

We further analyze whether the robustness failures exposed by HALLUCINATION-R1 reflect meaningful factual instability rather than trivial artifacts. Unless otherwise specified, all analyses use semantically consistent paraphrases generated by HALLUCINATION-R1 on the SimpleQuestions test set, with Qwen2.5-3B as the QA model.

6.1 Surface-Level Diagnostics under Semantic Control

We first test whether the observed failures can be explained by simple surface-level changes. As shown in Figure 2, robustness-failing and robustness-preserving paraphrases exhibit highly overlapping distributions in word count and dependency tree depth, suggesting that the failures cannot be reduced to longer or syntactically more complex

QA Model	Ori. Acc	Paraphrase Model	Consis@1	RF@1	RF@5	ER@1	WER@5
LLaMA-3.1-8B	247 (31.27%)	Qwen2.5-3B	64.68%	84	182	34.01%	73.68%
		Hallucination-R1 (Stage 1 Only)	80.37%	97	187	39.27%	75.71%
		Hallucination-R1	87.46% ^{↑22.78%}	136	215	55.06% ^{↑21.05%}	87.04% ^{↑13.36%}
Qwen2.5-3B	246 (31.14%)	Qwen2.5-3B	64.68%	81	183	32.93%	74.39%
		Hallucination-R1 (Stage 1 Only)	80.37%	99	193	40.24%	78.46%
		Hallucination-R1	87.46% ^{↑22.78%}	118	214	47.97% ^{↑15.04%}	86.99% ^{↑22.78%}
Qwen2.5-7B	323 (40.89%)	Qwen2.5-3B	64.68%	98	233	30.34%	72.14%
		Hallucination-R1 (Stage 1 Only)	80.37%	125	245	38.70%	75.85%
		Hallucination-R1	87.46% ^{↑12.60%}	147	277	45.51% ^{↑15.17%}	85.76% ^{↑13.62%}

Table 3: Ablation results of different paraphrase model variants across QA models on the TruthfulQA dataset. RF@1 denotes the number of robustness failures under a single semantically equivalent paraphrase, while RF@5 denotes the number of samples for which at least one out of five semantically equivalent paraphrases causes a robustness failure. $\uparrow / \downarrow$ denote the absolute difference (in percentage points) of each variant relative to the full HALLUCINATION-R1-3B, computed under the same QA model.

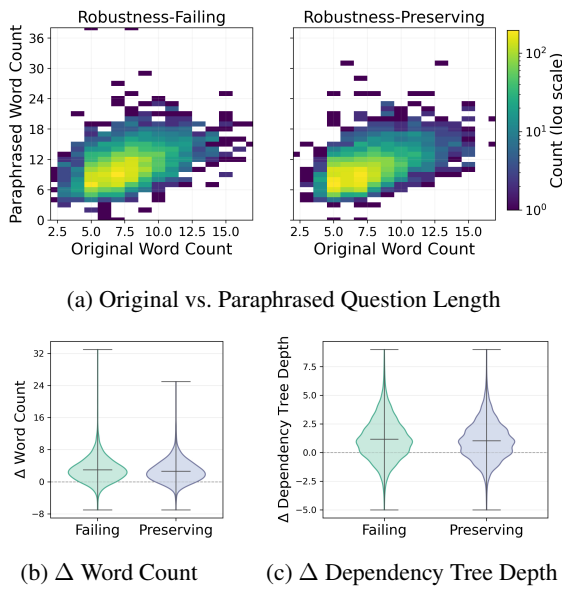


Figure 2: **Length and structural diagnostics of paraphrases.** (a) compares original and paraphrased question lengths for robustness-failing and robustness-preserving cases, denoted as Failing and Preserving in the plots. (b) and (c) report changes in word count and dependency tree depth, respectively.

questions. Additional normalized analyses are provided in Appendix G.1.

6.2 Question-Level Similarity versus Response-Level Drift

We next ask whether question-level semantic similarity is sufficient to ensure response-level factual stability. We compute L2-normalized embedding distances between original and paraphrased questions (d_Q) and between their question-answer concatenations (d_{QA}); details are given in Appendix G.2. As shown in Figure 3, robustness-failing paraphrases are associated with larger d_{QA} ,

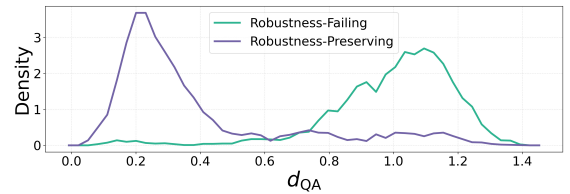


Figure 3: QA-concatenation embedding distance distributions for robustness-failing and robustness-preserving paraphrases.

whereas preserving paraphrases remain closer to the original QA representations. This suggests that semantic-preserving paraphrases can still induce response-level drift.

This drift is reflected inside the model. In the layer-wise heatmap of Figure 1, corresponding to the original and paraphrased questions, we conduct a logit-lens-style analysis and compare the log-probability difference between the first tokens of the correct and incorrect answers across layers. The original question increasingly favors the correct answer in later layers, whereas its semantically equivalent paraphrase shifts toward the incorrect answer. This study shows that the failure is reflected not only in the final response but also in late-stage factual preference dynamics. Additional definitions and examples are provided in Appendix G.3.

7 Robustness Training Utility

A central motivation of our framework is to construct paraphrase data that can serve as effective supervision for robustness-oriented training, rather than merely as probes for robustness evaluation. To examine this training utility, we conduct a lightweight SFT study by fine-tuning Qwen3-8B on semantically consistent and robustness-challenging

Paraphrase Model	QA Model	PopQA			SimpleQuestions		
		Para. Acc.	Robust Acc.	Any Acc.	Para. Acc.	Robust Acc.	Any Acc.
Hallucination-R1	Qwen3-8B	14.30%	3.79%	24.32%	21.17%	6.08%	38.33%
	Qwen3-8B-SFT	16.95% $\uparrow 2.65\%$	8.06% $\uparrow 4.27\%$	23.03% $\downarrow 1.29\%$	26.36% $\uparrow 5.19\%$	16.93% $\uparrow 10.85\%$	35.44% $\downarrow 2.89\%$
DeepSeek-V3	Qwen3-8B	15.90%	6.09%	26.47%	21.59%	5.72%	42.03%
	Qwen3-8B-SFT	17.88% $\uparrow 1.98\%$	10.62% $\uparrow 4.53\%$	24.49% $\downarrow 1.98\%$	26.26% $\uparrow 4.67\%$	18.33% $\uparrow 12.61\%$	35.96% $\downarrow 6.07\%$
Qwen2.5-32B	Qwen3-8B	16.70%	7.87%	26.16%	22.20%	6.33%	41.19%
	Qwen3-8B-SFT	18.06% $\uparrow 1.36\%$	11.14% $\uparrow 3.27\%$	24.06% $\downarrow 2.10\%$	26.70% $\uparrow 4.50\%$	18.72% $\uparrow 12.39\%$	35.91% $\downarrow 5.28\%$
Qwen3-8B	Qwen3-8B	17.09%	9.24%	24.52%	24.53%	10.84%	40.76%
	Qwen3-8B-SFT	18.70% $\uparrow 1.61\%$	12.87% $\uparrow 3.63\%$	23.03% $\downarrow 1.49\%$	27.42% $\uparrow 2.89\%$	19.11% $\uparrow 8.27\%$	35.90% $\downarrow 4.86\%$

Table 4: Robustness training results under semantics-preserving paraphrase variations. Para. Acc., Robust Acc., and Any Acc. are defined in Section 7. Paraphrase Model denotes the model used to generate evaluation paraphrases, while QA Model denotes the model being evaluated. $\uparrow / \downarrow$ denote absolute differences in percentage points between Qwen3-8B-SFT and Qwen3-8B under the same paraphrase model and dataset.

paraphrases generated by HALLUCINATION-R1 from SimpleQuestions. This experiment is intended to assess the utility of the generated data, rather than to propose a complete robustness training framework. Additional implementation details are provided in Appendix E.

For evaluation, each original question is paraphrased eight times by each evaluation paraphrase generator, and only semantically consistent paraphrases are retained. The evaluation paraphrase generators include HALLUCINATION-R1, DeepSeek-V3, Qwen2.5-32B, and Qwen3-8B, covering models with strong consistency–diversity trade-offs as well as a high-consistency but conservative paraphrasing baseline. Evaluation is conducted on SimpleQuestions and PopQA.

We report three metrics: Paraphrase Accuracy (Para. Acc.) is computed over individual paraphrased questions; Robust Accuracy (Robust Acc.) counts an original question as correct only if all of its semantically consistent paraphrases are answered correctly; and Any Accuracy (Any Acc.) counts it as correct if at least one paraphrase is answered correctly.

As shown in Table 4, fine-tuning with HALLUCINATION-R1-generated data improves Robust Acc., while Para. Acc. changes only modestly and Any Acc. consistently decreases. This pattern is consistent with robustness-oriented training: the model is not primarily encouraged to maximize occasional correctness on at least one variant, but to reduce failures across semantically equivalent variants. In other words, the model becomes better at maintaining factual consistency for questions it is already capable of answering, rather than merely acquiring additional factual knowledge.

Furthermore, the improvement in Robust Acc.

generalizes across both datasets and paraphrase models. The gains are not only observed on SimpleQuestions, from which the training data are constructed, but also transfer to PopQA. In addition, similar improvements are observed when evaluation paraphrases are generated by different paraphrase models, including HALLUCINATION-R1, DeepSeek-V3, Qwen2.5-32B, and Qwen3-8B. This indicates that the robustness improvement is not due to adapting to the specific linguistic patterns of HALLUCINATION-R1. Rather, HALLUCINATION-R1-generated data appears to capture broader paraphrase-induced fragility patterns shared across different paraphrasing distributions.

Overall, these results demonstrate the training utility of HALLUCINATION-R1-generated data. Semantically consistent yet challenging paraphrases provide effective robustness-oriented supervision by improving factual consistency under diverse meaning-preserving inputs, rather than primarily expanding the model’s factual knowledge.

8 Conclusion

We introduced HALLUCINATION-R1, a robustness-oriented paraphrase generation framework for factual consistency under semantic-preserving input variations. By combining semantic stabilization with robustness-challenging optimization, HALLUCINATION-R1 generates paraphrases that preserve factual intent while exposing robustness failures in downstream QA models. Experiments across multiple datasets and model families show that the generated paraphrases achieve a strong consistency–diversity trade-off, reveal non-trivial factual instability, and improve robust accuracy when used for lightweight fine-tuning. These re-

sults highlight robustness-oriented paraphrase generation as a direction for evaluating and improving factual consistency in large language models.

Limitations

This work has three main limitations. First, our evaluation relies on LLM-as-judge for semantic consistency and factual correctness assessment. This reflects a broader limitation in current factual QA and hallucination evaluation, where more authoritative and scalable factual verification signals are often unavailable. Although we conduct additional validation, judge errors or biases may still affect the results.

Second, our robustness-oriented SFT experiment remains preliminary. While the results suggest that semantically consistent yet challenging paraphrases can improve robustness under paraphrase variations, how to systematically use such data to strengthen models against paraphrase-induced factual fragility requires further study.

Third, the paraphrase distribution learned by HALLUCINATION-R1 is difficult to explicitly characterize. Our analyses show that the exposed failures cannot be explained by simple surface-level artifacts alone, but what linguistic or representational factors fundamentally make models vulnerable under semantic-preserving variations remains an open question.

Ethical Considerations

We use publicly available datasets and model checkpoints for research purposes, following their licenses and terms of use. The generated paraphrases, code, and models released with this work are intended for research on factual robustness evaluation and robustness-oriented training.

HALLUCINATION-R1 is designed to study and improve factual robustness under semantics-preserving paraphrase variations. The generated paraphrases are constrained to preserve the original factual query intent and are used in this work for controlled robustness evaluation and robustness-oriented training. The goal is not to create misleading user queries or promote factual errors, but to identify cases where models fail to maintain stable factual behavior under legitimate linguistic variation.

A potential risk is that robustness-challenging paraphrases could be misused to intentionally elicit incorrect answers from deployed models. We mit-

igate this risk by focusing on factual QA benchmarks with ground-truth answers, reporting aggregate robustness metrics rather than instructions for real-world misuse, and framing the method as a tool for diagnosis and robustness improvement. The paraphrases generated by our framework do not introduce new malicious content or harmful instructions; they reformulate existing factual questions while preserving meaning.

This work also uses automated LLM judges for semantic consistency and factual correctness evaluation. Such judges may introduce biases or errors, so their outputs should not be treated as definitive in high-stakes settings without additional verification. We therefore view HALLUCINATION-R1 as a research tool for analyzing and improving factual consistency, rather than as a standalone certification method for model reliability.

References

- Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Bowen Cao, Deng Cai, Zhisong Zhang, Yuexian Zou, and Wai Lam. 2024. On the worst prompt performance of large language models. *Advances in Neural Information Processing Systems*, 37:69022–69042.
- Minjoon Choi. 2025. Roparq: Paraphrase-aware alignment of large language models towards robustness to paraphrased questions. *arXiv preprint arXiv:2511.21568*.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.
- Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 968–988.
- Tingchen Fu and Fazl Barez. 2025. Same question, different words: A latent adversarial framework for prompt robustness. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31293–31307.

- Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 6065–6075.
- 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*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Kuan-Hao Huang, Varun Iyer, I-Hung Hsu, Anoop Kumar, Kai-Wei Chang, and Aram Galstyan. 2023. Paraamr: A large-scale syntactically diverse paraphrase dataset by amr back-translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8047–8061.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Riccardo Lunardi, Vincenzo Della Mea, Stefano Mizzaro, and Kevin Roitero. 2025. On robustness and reliability of benchmark-based evaluation of llms. *arXiv preprint arXiv:2509.04013*.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9802–9822.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. 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*, pages 9004–9017.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 856–865.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.
- 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*.
- Jianyi Zhang, Da-Cheng Juan, Cyrus Rashtchian, Chun-Sung Ferng, Heinrich Jiang, and Yiran Chen. 2024. Sled: Self logits evolution decoding for improving factuality in large language models. *Advances in Neural Information Processing Systems*, 37:5188–5209.
- Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

A Methodology Details

This appendix provides implementation details for the reward computation used in HALLUCINATION-R1. The overall objective and two-stage training pipeline are described in Section 3; here we focus on the concrete definitions of semantic consistency, group-relative diversity, and factual inconsistency exposure.

A.1 Semantic Consistency and Diversity Rewards

Given an original question x and a generated paraphrase y , we use DeepSeek-V3 to judge whether y preserves the semantic meaning of x . The semantic consistency reward is defined as:

$$R_{\text{consis}}(y) = \begin{cases} 1, & \text{if } M_{\text{consis}}(x, y) = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Semantic consistency also serves as a hard gate: semantically inconsistent paraphrases receive zero reward in both training stages.

To encourage diverse paraphrases under this constraint, we adopt a group-relative diversity reward. For each input x , we sample a group of paraphrases $\{y_1, \dots, y_G\}$ and retain the subset Y judged semantically consistent. For each $y \in Y$, we compute textual and semantic similarity scores. The textual similarity is based on Self-BLEU:

$$S_{\text{SelfBLEU}}(y) = -\frac{1}{G} \sum_{n=1}^G \text{SelfBLEU}_Y(y, n), \quad (5)$$

and the semantic similarity is based on cosine similarity between sentence embeddings:

$$S_{\text{embed}}(y) = -\sum_{y' \in Y} \frac{\phi(y) \cdot \phi(y')}{\|\phi(y)\|^2 \|\phi(y')\|^2}. \quad (6)$$

We rank candidates by the average similarity score $(S_{\text{SelfBLEU}} + S_{\text{embed}})/2$ in ascending order. The diversity reward is:

$$R_{\text{div}}(y) = \begin{cases} \frac{\text{Rank}(y, Y)}{|Y| - 1}, & \text{if } R_{\text{consis}}(y) = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The semantic stabilization reward follows the definition in Section 3:

$$R_{\text{stab}}(y) = R_{\text{consis}}(y) + \lambda_{\text{div}} R_{\text{div}}(y), \quad (8)$$

where λ_{div} is set to 1 in our experiments.

A.2 Factual Inconsistency Exposure Reward

For each semantically consistent paraphrase y , the QA model generates an answer $z \sim \pi_{\text{QA}}(\cdot | y)$. The LLM-as-judge M_{halluc} then evaluates the factual inconsistency between z and the ground-truth answer, producing a multi-level inconsistency score:

$$R_{\text{halluc}}(y) = M_{\text{halluc}}(y, z) \in \{0, 1, 2\}. \quad (9)$$

Here, higher values indicate stronger factual inconsistency. We use a three-level score rather than a binary correctness label to provide a more graded optimization signal, encouraging the generator to move toward paraphrases that more clearly expose factual robustness failures. The detailed judging prompt is provided in Appendix B.2.

The final training reward is aligned with Section 3:

$$R_{\text{train}}(y) = \begin{cases} R_{\text{stab}}(y) + R_{\text{halluc}}(y), & R_{\text{halluc}} \neq 0, \\ 0, & R_{\text{halluc}} = 0. \end{cases} \quad (10)$$

This design preserves semantic consistency as a hard constraint while using M_{halluc} as a factual inconsistency exposure signal.

A.3 Training Configuration

Model Setup We consider two distinct model roles during training: the QA model whose factual robustness is evaluated and the HALLUCINATION-R1 paraphrasing model. These two models are configured with different decoding strategies to reflect their functional objectives.

For the QA model, we disable stochastic decoding and adopt deterministic generation throughout all training and evaluation phases (i.e., `do_sample=False`). This design ensures that the QA model’s responses are stable and reproducible given a fixed input, eliminating randomness as a confounding factor when attributing factual errors to paraphrased queries. As a result, any observed factual failure can be reliably attributed to the meaning-preserving paraphrase rather than decoding noise.

In contrast, the HALLUCINATION-R1 model is trained and deployed with stochastic sampling enabled to encourage exploration of diverse paraphrasing strategies. Specifically, we adopt a sampling-based decoding configuration with `do_sample=True`, temperature-controlled generation, and nucleus sampling. The generation function is defined as:

Listing A.1: Hallucination-R1 configuration

```
do_sample = True
temperature = 0.7
max_tokens = 1024
top_k = 0
top_p = 0.95
```

This asymmetric decoding design is crucial: stochastic paraphrase generation enables broad exploration, while deterministic QA decoding ensures that observed factual errors arise from meaning-preserving input variations rather than QA output randomness.

A.4 Training Dynamics

Semantic Stabilization Training Results Figure 4 shows the reward curve during semantic stabilization. We train for one epoch on 1,536 samples, jointly optimizing semantic consistency and surface-level diversity. Although individual update steps exhibit variance due to GRPO sampling, the smoothed reward steadily increases from approximately 0.4 to above 1.2.

This trend indicates that the model gradually learns to preserve the original query meaning while exploring alternative syntactic and stylistic forms. It also provides a well-conditioned initialization for the subsequent robustness-challenging optimization stage.

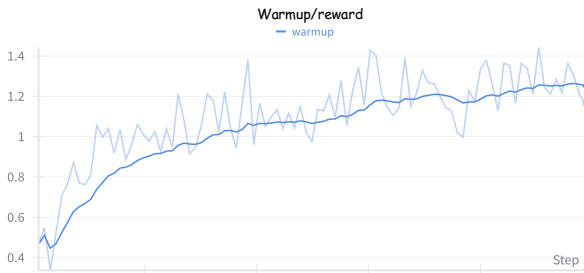


Figure 4: Training reward curve during the semantic stabilization stage.

Robustness-Challenging Optimization Dynamics After semantic stabilization, we conduct robustness-challenging optimization to maximize failure exposure under semantic invariance. As shown in Figure 5, the reward increases rapidly at the beginning and then enters a high-variance plateau, suggesting that the model first discovers basic failure-exposing patterns and then relies on

finer-grained interactions with the QA model’s decision boundaries.

Despite step-level variance, the smoothed reward remains stable around 2.5. The slight downward trend near the end suggests a trade-off between exploring more complex linguistic forms and maintaining semantic-preservation constraints. Overall, the second stage improves failure exposure while retaining controllability.

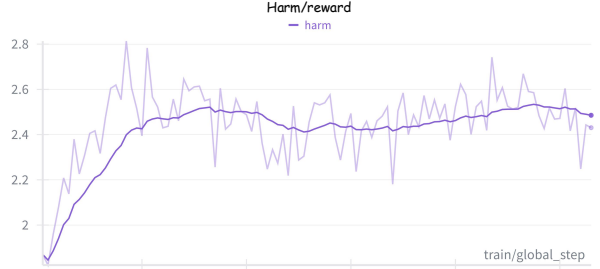


Figure 5: Training reward curve during the robustness-challenging optimization stage.

Training Infrastructure All training experiments were conducted on a single server equipped with 8 NVIDIA A800 80GB GPUs. We trained two separate HALLUCINATION-R1 models following the same training pipeline. For each model, the semantic stabilization stage required approximately 8 hours, and the robustness-challenging optimization stage required approximately 20 hours under the default configuration described above.

B Prompt Templates

B.1 Paraphrase Prompt

Listing B.1: Paraphrase Prompt

Your task is to paraphrase a factual question to expose weaknesses of the QA model under semantic invariance, while STRICTLY preserving the exact semantic meaning and intent of the original question.

CRITICAL CONSTRAINTS:

- Do NOT add any extra questions, conditions, or information.
- Do NOT change any key factual elements (tasks, entities, time, location, numbers, relationships, etc.).
- Do NOT change what is being asked — the answer to the paraphrased question must remain identical to the original.
- ONLY modify: word order, sentence structure, phrasing style, grammatical construction,

linguistic presentation.

Your output should include:

1. Verify the core semantic elements that **MUST** be preserved (what is being asked, about what/whom, under what conditions).
2. Identify stylistic transformation techniques that could make the question more challenging for factual QA **WITHOUT** changing meaning (e.g., passive voice, complex syntax, double negatives, nested clauses, unusual word order, formal/archaic language, indirect phrasing, pronoun substitution, nominalization).
3. Generate the paraphrased question that maintains semantic consistency but uses confusing stylistic patterns.

Output format:

<think> your analysis and paraphrasing strategy goes here </think>

<paraphrase> ONLY the paraphrased question, no explanations or analysis </paraphrase>

Original question: {question}

EVALUATION PROCESS:

1. Extract the core factual query that the paraphrased question is asking.
2. Check if any extra questions, conditions, or information have been added — if YES, result is 0.
3. Check if any key factual elements have been changed — if YES, result is 0.
4. Determine whether the original question's semantic intent is preserved — if YES, result is 1; otherwise result is 0.

Output format: "think": your process of thinking, "result": 1 or 0

Original question: question

paraphrased question: paraphrase

Output Only: "think": your process of thinking, "result": 1 or 0

B.2 Judge Prompt

Listing B.2: Consistency Judge Prompt

Please help me determine whether the paraphrased question generated by the paraphrase model has the same semantic meaning as the original question.

CRITERIA FOR SEMANTIC CONSISTENCY:

- The paraphrased question must **NOT** add any extra questions, conditions, or information beyond the original.
- The paraphrased question must **NOT** change any key factual elements (tasks, entities, time, location, numbers, relationships, etc.).
- The primary entities and query intent of the original question must be explicitly included in the paraphrased question (substituting similar words is acceptable).
- Modifying phrasing style, sentence structure, or word order is acceptable as long as the core question remains the same.
- The paraphrased question must **NOT** contain excessively long analysis, explanations, or irrelevant text beyond the question itself.

Listing B.3: Factual Inconsistency Judge Prompt

Your task is to evaluate whether the model's response correctly answers the question, based on the provided reference answer.

Use a 3-level score:

Score 0 (Exact Match): The response matches the reference answer exactly, or is semantically equivalent to the same entity.

Score 1 (Fuzzy Match): The response and the reference answer have a parent-child class relationship, or one is an abbreviated form of the other.

Score 2 (Mismatch): The response contradicts the reference answer, is factually wrong, or refers to a completely different entity.

This is part of an automated evaluation process. Therefore, the output must be a single JSON object in the following format: {"think": "your reasoning process", "result": 0/1/2}

Reference Answer: {ground_truth}

Model Response: {response}

Output only: {"think": "your reasoning process", "result": 0/1/2}

B.3 QA Model Answer Prompt

Listing B.4: QA Model Answer Prompt

You are a strict question answering model.
 You MUST output ONLY the final answer inside the <answer> and </answer> tags.
 Do NOT repeat the question.
 Do NOT explain.
 Do NOT output anything outside these tags.

C Experimental Setup

C.1 Paraphrase Generator Training Data

The training data for HALLUCINATION-R1 are derived from the training split of SimpleQuestions. To ensure that factual errors observed during training can be attributed to meaning-preserving paraphrase variations, we first run Qwen2.5-3B as the initial QA model (π_{QA}) on the training set and retain only instances that it answers correctly. After filtering, approximately 7,000 question-answer pairs remain for training the paraphrase generator π_{para} .

C.2 Paraphrase Generator Training Setup

We train two paraphrase generators based on Qwen2.5-1.5B and Qwen2.5-3B, respectively. Both models are optimized with the two-stage pipeline described in Section 3, yielding HALLUCINATION-R1-1.5B and HALLUCINATION-R1-3B.

Throughout the paper, we use HALLUCINATION-R1 to denote HALLUCINATION-R1-3B unless a specific model scale is explicitly stated.

C.3 Evaluation Datasets

We evaluate on SimpleQuestions, PopQA, and TruthfulQA. Table 5 summarizes the dataset sizes and evaluation protocols.

Dataset	Total Samples	Evaluation Metrics
SimpleQuestions	9,910	LLM Judge
PopQA	14,267	EM Match
TruthfulQA	790	LLM Judge

Table 5: Evaluation datasets and metrics.

C.4 Robustness Evaluation Protocol

Algorithm 1 summarizes the robustness evaluation procedure used to compute ER@1 and WER@5. A robustness failure is counted only when the paraphrased question is judged semantically equivalent

Algorithm 1 Robustness Evaluation under Semantic-Preserving Paraphrases

Require: Question x , ground-truth answer y

Ensure: Number of robustness failures ERR

```

1: ERR  $\leftarrow$  0
2:  $x' \leftarrow \pi_{para}(x)$ 
3: if  $M_{consis}(x, x') = True$  then
4:    $z \leftarrow \pi_{tgt}(x')$ 
5:   if  $M_{judge}(x, y, z) = False$  then
6:     ERR  $\leftarrow$  ERR + 1
7:   end if
8: end if
9: ErrorRate  $\leftarrow$  ERR/Num. of True Samples
10: return ERR, ErrorRate

```

to the original question and the QA model produces an incorrect answer.

C.5 Compared Paraphrase Models

In the consistency-diversity analysis, we compare HALLUCINATION-R1 with both open-source paraphrase generators and prompted LLM paraphrasing baselines. For trainable paraphrase generators, we include the Qwen2.5-1.5B and Qwen2.5-3B backbones, as well as their intermediate variants obtained after the Semantic Stabilization Stage. These models are used to examine how our two-stage training pipeline changes the paraphrase distribution relative to the base generators.

For prompted baselines, we compare against stronger instruction-tuned LLMs, including Qwen2.5-7B, Qwen2.5-32B (Qwen et al., 2025), Qwen3-4B (Yang et al., 2025), LLaMA-3.1-8B (Grattafiori et al., 2024), DeepSeek-V3, DeepSeek-R1 (Guo et al., 2025), and GPT-5 (Singh et al., 2025). All prompted baselines are evaluated using the same paraphrase-generation instruction and the same consistency/diversity metrics as HALLUCINATION-R1.

Unless otherwise specified, the QA models used in evaluation follow the same inference configuration as the QA model used for training feedback. We disable sampling for QA inference to ensure that observed factual errors arise from input-level paraphrase variation rather than output randomness. In contrast, paraphrase generation uses stochastic decoding to explore diverse paraphrases.

D Additional Robustness Evaluation Results

D.1 Effect of Model Scale and Training Strategy

To examine whether the ability to expose factual inconsistency comes primarily from model scale or from the proposed training strategy, we conduct additional comparisons on TruthfulQA using Qwen2.5-7B as the QA model. As shown in Table 6, HALLUCINATION-R1-3B achieves stronger ER@1 and WER@5 than larger untrained paraphrase models, including Qwen2.5-7B and Qwen2.5-32B, while maintaining high semantic consistency. Compared with the Stage-1-only variant, the full model also yields substantially higher failure exposure, showing the contribution of the robustness-challenging optimization stage. These results suggest that the effectiveness of HALLUCINATION-R1 stems from the proposed training strategy rather than parameter scale alone.

E Robustness-Oriented SFT Details

This appendix provides implementation details for the robustness-oriented SFT experiment in Section 7.

E.1 SFT Data Construction

The SFT data are constructed directly from the training split of SimpleQuestions, without filtering original questions based on QA model correctness. For each original question, HALLUCINATION-R1 generates multiple robustness-challenging paraphrases. We retain only paraphrases that pass the semantic consistency verifier and pair each retained paraphrase with the original ground-truth answer. This construction provides robustness-oriented supervision over meaning-preserving surface variations while relying only on the original dataset annotations.

E.2 SFT Training Setup

We fine-tune Qwen3-8B with LoRA-based supervised fine-tuning. The objective is the standard next-token prediction loss over the ground-truth answer, conditioned on robustness-challenging paraphrased questions. Only LoRA parameters are updated, while the base model remains frozen.

E.3 SFT Evaluation Protocol

For evaluation, we follow Section 7: each original question is paraphrased eight times by each

paraphrase model, and only semantically consistent paraphrases are retained. We report Paraphrase Acc., Any Acc., and Robust Acc. Robust Acc. is computed at the original-question level and requires all retained paraphrases of the same original question to be answered correctly.

F Judge Consistency Validation

F.1 Cross-Model Consistency Validation

To verify that semantic consistency evaluation does not depend on a single judge model, we conduct a cross-model validation experiment with LLM judges of different architectures. Each judge independently evaluates the paraphrased samples and produces binary consistency decisions. We report **Consis@1**, **Consis@5**, and **Majority@5**, following the definitions in the main paper.

As shown in Table 7, the consistency judgments are highly aligned across judge models. This suggests that the semantic consistency evaluation is robust to the choice of judge model and reflects shared agreement across different LLMs.

F.2 Human Consistency Annotation Study

To further rule out shared biases among LLM judges, we conduct a small-scale human annotation study. We randomly sample 100 original questions from the TruthfulQA paraphrased set, yielding 500 question–paraphrase pairs. Each pair is annotated by at least three graduate students with NLP backgrounds recruited for a small-scale expert validation study. No monetary compensation was provided, and only aggregate annotation results are reported.

Annotators only see the original and paraphrased questions. They were instructed: “Given an original question and a paraphrased question, decide whether the paraphrase preserves the original querying intent and factual reference. Mark the pair as consistent only if it does not add, remove, or change key entities, relations, conditions, or the answer-bearing intent.” The final label is determined by majority voting.

As shown in Table 7, human judgments are broadly aligned with LLM-based consistency judgments. This provides additional evidence that the semantic consistency evaluation is not solely driven by shared LLM judge biases, but is also consistent with human intuition.

QA Model	Ori. Acc	Paraphrase Model	Consis@1	RF@1	RF@5	ER@1	WER@5
Qwen2.5-7B	323 (40.89%)	Qwen2.5-3B	64.68%	98	233	30.34%	72.14%
		Qwen2.5-7B	77.32%	124	262	38.39%	81.11%
		Qwen2.5-32B	93.17%	135	256	41.80%	79.26%
		DeepSeek-V3	97.72%	164	273	50.77%	84.52%
		Hallucination-R1 (Stage 1 Only)	80.37%	125	245	38.70%	75.85%
		Hallucination-R1-3B	87.46%	147	277	45.51%	85.76%

Table 6: Additional results on TruthfulQA with Qwen2.5-7B as the QA model. We compare HALLUCINATION-R1-3B with paraphrase models of different scales and architectures.

Judge Model	Consis@1	Consis@5	Majority@5
DeepSeek-V3 (Liu et al., 2024)	87.46%	99.87%	97.09%
Kimi-K2 (Guo et al., 2025)	83.72%	99.49%	95.87%
Qwen3-235B-A22B (Yang et al., 2025)	85.82%	100.00%	96.46%
Human (100 samples)	83.00%	98.00%	96.00%

Table 7: Semantic consistency evaluation on the TruthfulQA dataset across different judge models and human annotation.

G Supplementary Analysis

This section provides supplementary analyses for Section 6. We expand the surface-level diagnostics, embedding-distance analysis, and layer-wise factual preference analysis used to examine whether robustness failures exposed by HALLUCINATION-R1 can be attributed to trivial artifacts or reflect non-trivial factual instability.

G.1 Length and Structure Features

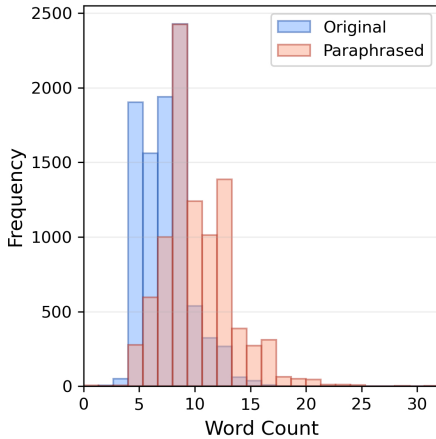


Figure 6: Distribution of word counts for original and paraphrased questions across all questions.

Sentence Length Distributions. As shown in Figure 6, HALLUCINATION-R1 exhibits a clear tendency to increase sentence length. However, sentence length alone does not clearly distinguish robustness-failing from robustness-preserving paraphrases.

Dependency Tree Depth. Dependency tree depth in Figure 2c is computed using an off-the-shelf parser based on the Universal Dependencies (UD) formalism. We define tree depth as the maximum distance from the root token to any node in the dependency tree. Our analysis focuses exclusively on relative depth changes (Δ) induced by paraphrasing, rather than absolute depth values.

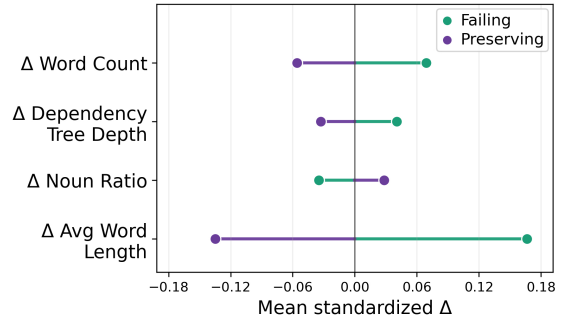


Figure 7: Normalized feature profiles of robustness-failing and robustness-preserving paraphrases under the global distribution.

Normalized Structural Feature Profiles. Figure 7 presents normalized feature profiles for robustness-failing and robustness-preserving paraphrases under a global distributional reference.

We refer to paraphrases that lead the QA model to incorrect answers as robustness-failing (RF) paraphrases, and those that preserve correct QA behavior as robustness-preserving (RP) paraphrases.

Feature definitions. Surface length and lexical form complexity are treated as complementary signals. Changes in sentence length are measured by the word-count difference:

$$\Delta WC = |\mathcal{W}^{\text{para}}| - |\mathcal{W}^{\text{orig}}|, \quad (11)$$

where $|\mathcal{W}|$ denotes the number of words in a question. In contrast, changes in average word length

capture shifts in lexical morphology (e.g., preferences for longer or more complex word forms):

$$\Delta\text{AWL} = \frac{|\mathcal{C}^{\text{para}}|}{|\mathcal{W}^{\text{para}}|} - \frac{|\mathcal{C}^{\text{orig}}|}{|\mathcal{W}^{\text{orig}}|}, \quad (12)$$

where $|\mathcal{C}|$ denotes the number of characters. While ΔWC reflects the external scale of sentence length (information volume), ΔAWL isolates changes in word-form complexity independent of sentence length.

Normalization procedure. For each feature dimension, standardized values are computed with respect to the overall distribution across all paraphrases. Let Δf_i denote the change value of a structural feature (e.g., word count, dependency tree depth, noun ratio) for sample i . The global mean and standard deviation are defined as:

$$\mu_f = \frac{1}{N} \sum_{i=1}^N \Delta f_i, \quad (13)$$

$$\sigma_f = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta f_i - \mu_f)^2}, \quad (14)$$

where N denotes the total number of labeled samples.

Each sample is then mapped to a standardized score:

$$z_i = \frac{\Delta f_i - \mu_f}{\sigma_f}. \quad (15)$$

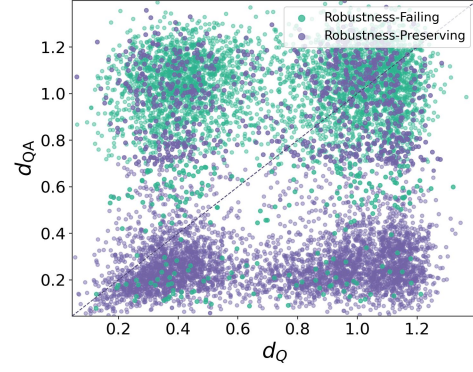
For visualization, mean standardized scores are reported separately for robustness-failing and robustness-preserving paraphrases:

$$\bar{z}_{\text{RF}} = \mathbb{E}[z_i \mid i \in \text{RF}], \quad \bar{z}_{\text{RP}} = \mathbb{E}[z_i \mid i \in \text{RP}]. \quad (16)$$

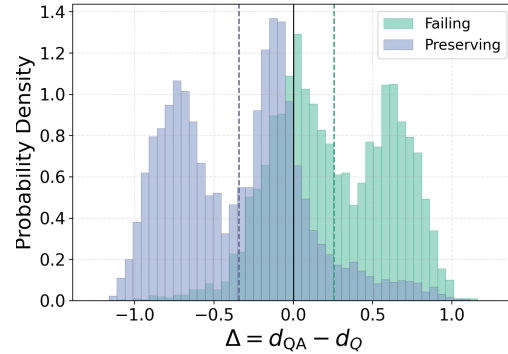
Although absolute feature distributions largely overlap, normalization reveals consistent directional shifts in group-wise means. In particular, robustness-failing paraphrases exhibit small but systematic tendencies across multiple dimensions, including slightly increased sentence length, moderately higher syntactic complexity, and a reduced inclination toward nominalization.

G.2 Embedding Distance

We detail the embedding distance computation. Both original and paraphrased questions, as well as their corresponding generated answers, are formatted using the following templates:



(a) Joint distribution of d_Q and d_{QA}



(b) Distribution of Δ

Figure 8: **Relationship between question paraphrase distance and QA-concatenation distance.** Robustness-failing and robustness-preserving paraphrases exhibit distinct patterns in both the joint distance space and the induced Δ distribution.

Listing G.1: Embedding Input Templates

<i>Question:</i>	"Question: {question}"
<i>QA-concatenation:</i>	"Question: {question}\n " "Answer: {answer}"

All formatted texts are fed into Qwen2.5-3B. Hidden states are extracted from the 18th transformer layer (out of 36 total layers), and the hidden state of the final token is used as the sequence embedding. Pairwise Euclidean (L2) distances between embeddings are computed and normalized.

Based on the above distance definitions, we further define a relative distance difference:

$$\Delta = d_{QA} - d_Q. \quad (17)$$

A positive Δ indicates that the generated response amplifies the semantic deviation introduced by paraphrasing, whereas a negative Δ suggests that the response pulls the joint QA representation back toward the original semantic neighborhood.

Under this definition, robustness-failing paraphrases predominantly lie in regions with positive Δ , while robustness-preserving paraphrases cluster around negative or near-zero Δ , a pattern further confirmed by the Δ histogram (Figure 8).

G.3 Layer-wise Factual Preference Dynamics

We provide additional details for the layer-wise case study shown in Figure 1 and present further examples in Figure 9. We conduct a logit-lens-style analysis by projecting the hidden state at each layer to the vocabulary space using the model’s output head. This allows us to inspect how the model’s preference between competing answer tokens evolves across layers when a semantic-preserving paraphrase changes the model’s response from correct to incorrect.

For each case, we compare the layer-wise log-probability difference between the first token of the correct answer and that of the incorrect answer generated under the paraphrased input. Formally, for an input question q and each layer ℓ , we compute:

$$\Delta \log p^{(\ell)} = \log p^{(\ell)}(t_{\text{gold}} | q) - \log p^{(\ell)}(t_{\text{err}} | q), \quad (18)$$

where t_{gold} denotes the first token of the ground-truth answer and t_{err} denotes the first token of the incorrect answer generated under the paraphrased input. A larger value indicates a stronger preference for the correct answer token, while a smaller value indicates a shift toward the incorrect one; accordingly, redder regions in the visualization indicate stronger correct-answer preference, and bluer regions indicate stronger incorrect-answer preference. We focus on the first answer token to capture the model’s initial factual preference and avoid confounding effects from downstream autoregressive dependencies.

Figure 9 presents three representative factual inconsistency cases. Panels (a) and (b) correspond to two different paraphrases of the same original question, where the ground-truth answer is *Greek*. Although the two paraphrases lead to different incorrect answers, namely *film* and *English*, they exhibit a similar pattern: the original question develops a stronger preference for the correct answer in later layers, while the paraphrased question shifts the late-layer preference toward the incorrect answer. Panel (c) shows a different type of error, where the correct answer *spanish* is replaced by an *unknown* response under the paraphrased input.

Across these cases, preference differences be-

tween original and paraphrased inputs remain relatively small in early and middle layers, but become more pronounced in later layers. This suggests that robustness failures under semantic-preserving paraphrases are associated with late-stage factual preference shifts, rather than a complete disruption of early lexical or semantic representations.

These observations align with prior layer-wise analyses showing that vocabulary projections can reveal evolving factual preferences across layers, and that later layers are closely related to factual prediction and answer calibration (Chuang et al., 2023; Zhang et al., 2024). We therefore use the heatmaps as diagnostic evidence that paraphrase-induced failures are accompanied by late-stage factual preference shifts, rather than as causal proof of the failure mechanism.

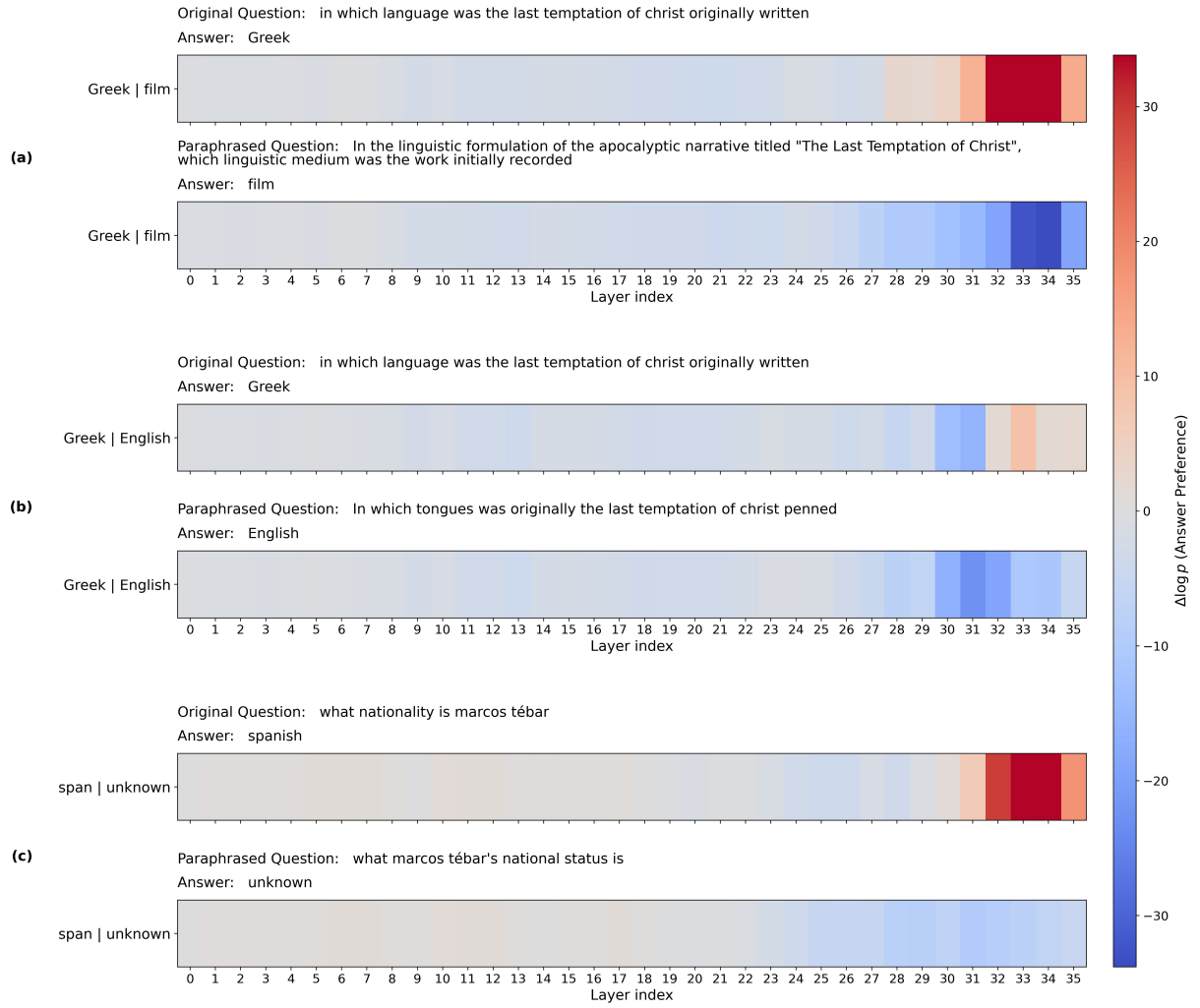


Figure 9: Additional examples of layer-wise factual preference dynamics under original and paraphrased inputs.