

Do Not Score the Edited Answer: A Negative Result on Loose Evaluation for Verifiable Instruction Following

José Luis Delgado

Universitat Oberta de Catalunya

Barcelona, Spain

jdelgado13@uoc.edu

Abstract

Verifiable instruction-following benchmarks often include strict and loose scoring modes: strict scoring checks the original response, and loose scoring applies the checker to transformed versions of the response, for example after removing markdown or dropping boundary lines, to recover false negatives caused by harmless wrappers and formatting artifacts. We study a negative result for this repair strategy using a constructive suite of 144 canaries in which every valid wrapper is paired with an invalid deletion trap that has the same transformed-response checker outcomes under IFEval-style transformations. Loose scoring accepts all valid wrappers and all invalid traps; support over transformation functions accepts both classes, and a threshold that deduplicates passing edited strings rejects both classes. On the balanced suite, any scorer that observes only the transformed-view pass signature has a best possible accuracy of 0.50, and the canaries are solved only by a boundary-aware certificate that inspects the original deleted span. Deletion- or normalization-based loose evaluation therefore needs evidence-preservation checks when edited spans may affect instruction satisfaction.

1 Introduction

Verifiable instruction-following evaluation uses deterministic checkers to replace open-ended judgment with executable constraints. IFEval is the canonical example: each prompt contains one or more verifiable instructions, and each instruction is scored by a corresponding verification function (Zhou et al., 2023). FollowBench and IFBench extend the same line of work through richer constraints and larger instruction inventories (Jiang et al., 2024; Pyatkin et al., 2025). These benchmarks reduce dependence on preference modeling, although their scores still depend on the response object supplied to the checker.

The strict and loose modes used by IFEval expose this design choice directly. Strict scoring applies a checker to the original model response, whereas loose scoring applies the checker to several transformed response views, such as versions with markdown removed or the first or last line dropped, and accepts the response when any transformed view passes. Loose scoring can repair cases where an otherwise correct answer is surrounded by boilerplate or formatting text that causes a brittle exact checker to fail.

The same strategy can also delete evidence that determines whether the original response satisfies the instruction. If the removed line is a harmless wrapper, loose scoring repairs a false negative; if the removed line violates the instruction, loose scoring evaluates an edited answer that no longer contains the violation. The difference is semantic, but the loose criterion only observes checker outcomes on transformed strings.

Zhou et al. note that loose scoring can introduce false positives (Zhou et al., 2023). The canaries formalize that observation as an indistinguishability condition: for deletion-based loose scoring, a valid repair and an invalid deletion trap can have the same transformed-view pass signature, so no rule based only on that signature can separate them.

The self-contained canary suite specifies an answer line and a boundary policy. Boundary lines may appear before, after, or around the answer line, but they must not violate a stated policy such as containing a forbidden word, comma, or digit. For each setting, we construct a valid wrapper and an invalid deletion trap with the same target answer and the same transformed-view checker outcomes. The valid case contains a harmless boundary line, and the invalid case contains a boundary line that violates the policy. After the relevant boundary line is dropped, both reduce to the same answer line.

The suite isolates transformed-response scoring from model behavior: the exact-checker strict base-

line rejects all canaries because the original response includes boundary text, while loose scoring accepts all valid wrappers and all invalid deletion traps. A support-count filter that requires at least two passing transformed views accepts the same cases, because the valid and invalid members of each pair have identical passing-view signatures. More generally, any classifier whose evidence is limited to the transformed-view pass signature, the target, the boundary position, and the instruction policy has a best possible in-sample accuracy of 0.50 on the balanced suite.

Comparing signature-only rules with a simple boundary certificate identifies the missing evidence. The certificate accepts a transformed-view pass only when the deleted boundary span does not violate the original policy, and it solves the suite by using the original response span removed by the transformation. As a diagnostic for this canary family, the certificate shows that loose scoring by edited responses requires an evidence-preservation check when the edit can delete instruction-relevant text.

2 Loose Scoring by Response Transformation

Let I be an instruction and y a model response. The predicate $V_I(y)$ denotes validity of the original response under the full instruction, including any boundary or formatting condition, while $e_I(x)$ denotes a deterministic verification function applied to a particular string x . The checker may be brittle with respect to wrapper text. Strict scoring applies the checker to the original response,

$$z_{\text{strict}}(I, y) = e_I(y),$$

whereas loose scoring accepts if any transformed response passes,

$$z_{\text{loose}}(I, y) = \max_{t \in T} e_I(t(y)),$$

where T is a set of response transformations. This form covers the standard loose-evaluation pattern in which wrappers or formatting artifacts are removed before checking. The canary result concerns scorers that approximate $V_I(y)$ using only checker outcomes $e_I(t(y))$ over transformed response views.

For a fixed I , define the transformed-view pass signature

$$\sigma_I(y) = (e_I(t(y)))_{t \in T}.$$

The support count is $\sum_{t \in T} e_I(t(y))$. A support-count filter is still a function of $\sigma_I(y)$; it can require more than one passing view, but it cannot inspect content that all relevant passing transformations removed.

Indistinguishability. Suppose two responses y^+ and y^- satisfy $V_I(y^+) = 1$, $V_I(y^-) = 0$, and $\sigma_I(y^+) = \sigma_I(y^-)$. Any scoring rule $g(I, \sigma_I(y))$ assigns the same verdict to both responses and must be wrong on at least one of them. If a canary suite is balanced into valid and invalid pairs with identical signatures, the best possible accuracy for any such rule is 0.50. The bound follows from the information available to the scorer and does not depend on model behavior.

3 Constructive Canaries

The suite contains 144 canaries obtained by crossing six target answers, three boundary positions, and four boundary policies, with one valid wrapper and one invalid deletion trap for each setting. The target answers are blue, Yes, done, cat, Paris, and 42. Boundary lines may appear as a prefix, suffix, or both, and the policies forbid one of four surface properties: the word red, the word banana, a comma, or a digit.

Each instruction has the same form: the answer line should be exactly the target, and boundary lines may appear in the specified position but must not violate the stated policy. The checker is brittle by construction: it passes only when the checked string is exactly the target answer. The transformation set contains the eight IFEval-style transformations used in the experiment: identity, markdown removal, first-line deletion, last-line deletion, first-and-last-line deletion, and the corresponding markdown-removal compositions. Because paired cases differ only in whether the deleted boundary line violates the policy, the suite isolates the scoring issue from response generation.

The canaries are synthetic, but their ingredients match ordinary verifiable-instruction components: exact constrained answers, forbidden tokens, punctuation constraints, and beginning, ending, or whole-response conditions. IFEval-style loose transformations explicitly remove first or last lines to skip boilerplate, which is the operation used by the deletion traps.

In the representative pair in Table 1, the valid wrapper is acceptable under the original instruction because answer note does not violate the red

Case	Original response	After deletion
Valid wrapper	answer note	
blue	blue	
Deletion trap	red note	
blue	blue	

Table 1: Representative canary pair for an instruction that requires the answer line to be exactly blue and allows a prefix line only if it does not contain red; dropping the first line makes both responses identical to the checker.

Method	Valid recall	Invalid accept	Acc.
Exact strict	0.00	0.00	0.50
Loose edited-view	1.00	1.00	0.50
Support ≥ 2	1.00	1.00	0.50
Unique support ≥ 2	0.00	0.00	0.50
Boundary certificate	1.00	0.00	1.00

Table 2: Synthetic canary results for exact strict scoring, loose edited-view scoring, support thresholds, and the boundary certificate; only the boundary certificate inspects the original deleted boundary span.

policy, while the deletion trap is invalid because red note violates the policy. A first-line deletion transformation maps both responses to blue, so the transformed response contains no evidence about the boundary violation.

4 Results

The main canary results in Table 2 show that, under the brittle exact-answer checker, strict scoring has zero valid recall because every original response contains an allowed or disallowed boundary line and the checker only accepts the bare answer. Loose edited-view scoring reaches perfect recall on valid wrappers and accepts every invalid deletion trap for the same reason. Requiring at least two passing transformation functions does not change the outcome, since the paired valid and invalid cases have the same passing transformations.

The unique-support row deduplicates passing transformed strings before counting support. It is not a content-aware scorer: it ignores the contents of failing views, including the original response. Under this rule, each canary has one unique passing string, the bare target, so a threshold of two rejects both valid wrappers and invalid traps, yielding the same 0.50 accuracy by trading false positives for false negatives.

We report the unique-support variant as a support-threshold heuristic, separate from the

Quantity	Value	Share
Signature groups	72	–
Canary cases	144	1.00
Minimum signature errors	72	0.50
Upper-bound accuracy	–	0.50

Table 3: Best possible in-sample accuracy for any classifier that observes only the transformed-view pass signature, target, deleted-span position, and instruction policy; mixed-signature groups force at least one error per valid/invalid pair.

Deleted span	Policy	Valid	Invalid
both	digit	6	6
prefix	comma	6	6
prefix	red	6	6
suffix	banana	6	6
All signatures	all	72	72

Table 4: Examples of indistinguishable transformed-view signatures; each row aggregates six target-specific mixed groups in which the valid wrapper and invalid deletion trap share the same target, deleted-span position, policy, and passing transformed views.

signature-only upper bound in Table 3. A content-aware scorer that inspects the original response is outside that evidence restriction and corresponds to the boundary certificate.

In the boundary-certificate row, the scorer solves the suite by inspecting the original deleted span. The certificate first requires a transformed-view pass, identifies the boundary span deleted by the passing transformation, checks that span against the original policy, and accepts only if the deleted span does not violate the policy. It accepts every valid wrapper and rejects every invalid trap, which identifies whether the transformation removed instruction-relevant evidence as the missing information.

The corresponding indistinguishability bound in Table 3 follows from 72 mixed signature groups, each containing one valid wrapper and one invalid deletion trap with the same target, deleted-span position, instruction policy, and transformed-view pass signature. Any classifier restricted to that evidence assigns the same label to both members of each pair, producing at least 72 errors on the balanced suite.

The grouped signatures in Table 4 aggregate over target answers within a fixed deleted-span position and policy. The total row covers all mixed groups, where the view-signature evidence is exactly bal-

anced between valid and invalid cases. Additional passing views therefore do not constitute semantic confirmation; passing views only say that some edited response satisfies the checker.

5 Discussion

The negative result applies to loose evaluation schemes that score edited responses without certifying that the edit preserves evidence needed by the original instruction, including support-count variants that coarsen the transformed-view signature. The construction does not require model-specific adversarial behavior; it applies whenever a model output contains instruction-relevant boundary text that a loose transformation deletes. We do not estimate how often such traps occur in deployed outputs, since the result is an information-loss canary with no prevalence estimate.

For response transformations used to repair brittle checkers, the boundary certificate gives the relevant evidence-preservation condition: the transformation should be paired with a certificate over the original response span. For boundary deletion, that certificate must check whether the removed span is irrelevant to the instruction being scored. For formatting normalization, it must check whether the normalized surface preserves the property under evaluation. Without that evidence-preservation check, loose scoring measures an edited answer and not the model response that was actually produced.

Scope beyond deletion. The indistinguishability argument is not specific to line deletion. It applies to any many-to-one transformation for which a compliant output and a violating output map to the same transformed view and therefore induce the same checker signature. Markdown stripping, case folding, whitespace normalization, and punctuation removal inherit the same impossibility whenever the erased distinction is instruction-relevant. Conversely, an injective normalization over the relevant output space does not create this particular information-loss failure. The risk is therefore tied to lossy transformations, rather than to normalization per se.

The canary construction aligns with prior work on evaluation sensitivity to formatting and surface variation (Sclar et al., 2024). It follows the logic of behavioral testing by isolating a failure mode of the evaluation procedure, with no estimate of failure rates for any particular model distribution (Ribeiro et al., 2020).

6 Conclusion

Loose scoring by response transformation is reliable as a repair only when the transformation preserves the evidence needed by the original instruction. The canary suite gives an exact failure case for deletion-based loose scoring: each valid wrapper and its invalid deletion trap have the same target answer and the same transformed-view pass signature after deletion, so any signature-only scorer is limited to 0.50 accuracy on the balanced suite. Counting more passing transformations does not add semantic evidence, and deduplicating passing edited strings changes the error type without recovering the deleted span.

By solving the suite only after reading the original span removed by the transformation, the boundary certificate identifies the missing variable. For verifiable instruction following, a loose score obtained from an edited response should be interpreted as a score for that edited response unless the scorer also certifies that the edit preserved instruction-relevant evidence. The canaries turn this requirement into a testable condition for deletion- and normalization-based repairs.

Limitations

The canaries are synthetic and balanced by construction, so they establish an information-loss failure mode without estimating its empirical frequency for deployed model outputs. The suite tests exact-answer constraints with boundary policies over prefix and suffix text; other instruction families used in verifiable instruction-following benchmarks would require additional canary constructions. The boundary certificate is included as a diagnostic for these canaries, and a production evaluator for broader constraint families would require instruction-specific evidence-preservation rules.

Our balanced synthetic canaries establish possibility and an information-theoretic limitation; they do not estimate the prevalence of deletion traps in naturally generated model outputs. Estimating that prevalence would require a separately sampled corpus of raw outputs and human adjudication of whether discarded boundary material violates the original instruction. Accordingly, the 0.50 bound should not be interpreted as an estimate of the benchmark-wide false-positive rate. It characterizes the worst-case identifiability of evaluators restricted to transformed-view signatures.

Ethical Considerations

The study uses synthetic text generated by deterministic scripts and does not involve human subjects, private data, or model-generated personal information. The main risk is misuse of the result as a claim that loose scoring is always invalid.

References

- Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. [Follow-Bench: A multi-level fine-grained constraints following benchmark for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. 2025. [Generalizing verifiable instruction following](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- 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, Online. Association for Computational Linguistics.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting](#). In *The Twelfth International Conference on Learning Representations*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.