
Prompts Are Specifications: Verifying Commitment Fidelity in LLM Outputs

Eduardo Perez Denadai
Optimality.ai

Abstract

Standard correctness toolchains—compilers, linters, and test suites—evaluate code validity without reference to the original prompt specifications, allowing unfaithful modifications to pass undetected: a rename that drops every call site parses, type-checks, lints, and passes the tests. Code-editing models are replicated at 25% silent wrongness: parse-valid, unfaithful, compiler-blind—and the rate falls with file size, because bigger files hide smaller diffs. We call the missing property *commitment fidelity*—does the response still honour what the prompt committed to?—and build the verifier the property deserves, on one contract: *a check is entitled to exactly as much authority as the decidability of its segment*. Three gates are exactly decidable and fire hard: a unit swapped (30 seconds $\rightarrow$ 30 minutes—invisible to compilers, tests, and numeric diffs), a unit dropped while the number survives, and a forbidden token added in the diff. Two advisory checks measure conservation of committed entities and numbers without blocking; an ambiguity detector answers ASK instead of guessing when the prompt itself is under-determined; everything else is recorded *unverifiable*, never failed. On 14 labelled cases, both directions scored: 13/14 with one documented miss, zero unexpected passes. On a 113-turn real transcript, gates fire on 3% of turns and advisories on 30%—the gate/advisory boundary was *measured*, not chosen, and a scoping ablation (50% $\rightarrow$ 30% affected turns) plus a rejected copula extension (+89% noise to recover one case) show exactly how. Comparative experiments rule out the obvious alternatives: a keyword/regex baseline scores *below* the empirical chance floor (0.090–0.117 vs 0.1228) on real commits; a $15\times$ larger encoder buys accuracy, not a different capability (and a similarity-trained encoder is worst of three); a learned commitment classifier is reported inconclusive at current label counts rather than shipped as a gate. Exit codes are the API—`exit 0` build, `exit 1` blocked, `exit 2` ask, `exit 3` usage, never a scalar—ensuring that non-decidable checks never emit misleading confidence metrics that would create a false perception of verification coverage.

1 Introduction

A prompt is a specification. Someone—a user, a ticket, an agent instruction—commits to things in it: *30 seconds*, *without retries*, *parse_config in engine.py*. Existing tooling verifies *correctness*: does the artifact work? Nothing verifies *fidelity*: does the artifact still honour the commitments? The stakes are measurable: code-editing models are replicated at 25% silent wrongness—parse-valid, unfaithful, compiler-blind—and the rate *falls* with file size, because bigger files hide smaller diffs. The silent 25% is a fidelity failure wearing a correctness costume.

The fidelity checker answers one question: *does the response still honour what the prompt committed to?* Correctness is a predicate of the artifact against its environment; commitment fidelity is a

predicate of the response against its prompt—independent properties, and only the second is absent from the toolchain. The design contract is stated once and enforced everywhere:

Every check is entitled to exactly as much authority as the decidability of its segment. Checks that cannot decide say unverifiable—recorded, never failed. A check that lies about what it can decide is worse than no check: it emits confidence where no decision exists.

Method chain. prompt $\rightarrow$ inferred intent $\rightarrow$ implementation $\rightarrow$ response/diff $\rightarrow$ gate: a chain of four segments, each with its own decidability class. The first segment is largely exact (promises carry concrete checkable claims); the middle segment—is this response a faithful implementation?—is semantic equivalence, undecidable in general; the whole chain is not decidable at all. So the instrument does three honest things: gates the exactly decidable slice, *measures* the noisy slice with a scoping discipline, and *attributes* every caught failure to the link that broke—leaving the residual undecidable slice quantified rather than hidden.

Contributions: (1) the fidelity formulation with per-segment decidability and the authority contract (§3); (2) five checks with deliberately classified authority—gate, advisory, unverifiable, and an ASK intervention—plus an exit-code contract that makes the authority limit enforceable (§4); (3) an evaluation protocol in which the gate/advisory boundary is measured on real transcripts and every scoping decision is an ablation (§5); (4) negative results as method: regex below the chance floor, encoder capacity not being the axis, and an inconclusive learned classifier reported rather than shipped (§6).

2 Related work

Similarity and judging. BLEU-style overlap [Papineni et al., 2002] and embedding similarity (including models trained specifically for sentence similarity [Gao et al., 2021]) score topical agreement, not commitment: a unit swap conserves the number and destroys the meaning, so any scalar over numbers passes it. LLM-as-judge is the modern default and a documented failure regime—position, verbosity, and self-enhancement biases [Zheng et al., 2023], reward hacking under specification mismatch [Skalse et al., 2023]; judge-free alternatives such as AI feedback [Lee et al., 2023] still ground rewards in prose. Our gates compute exact predicates over parsed forms and never consult a model.

Instruction following and faithfulness. Instruction-following benchmarks [Zhou et al., 2023] test whether outputs satisfy verifiable constraints at generation time; we verify the *response against its own prompt* after the fact, for the commitments that prompt actually made—a subset relation in the other direction, and the only one that can run on arbitrary existing traffic.

Requirements traceability. Classical traceability asks whether each requirement survives into the artifact [Gotel and Finkelstein, 1994]; document-level referential-integrity checkers resolve citations to sections. This checker extends traceability from document $\leftrightarrow$ document to prompt $\leftrightarrow$ response—*commitment traceability*—with the same three-valued outcome (resolved, violated, unresolved) and the same discipline: read the evidence, not the count.

3 Formalization

Definition 1 (Commitment). *A commitment is a concrete claim inside a directive sentence of the prompt: a number (unit-bearing or bare), a named artifact (backtick span, file path, or snake-case identifier), or a concretely negated token (without retries, never cache). Mentions, asides, and pasted data are not commitments.*

Definition 2 (Verdict). *BUILD: every committed claim survived. ASK: the prompt itself is under-determined—a question for the person who asked. BLOCKED: a committed claim was violated. Advisory findings never block; unverifiable findings are never counted as failures.*

Proposition 1 (Authority bound). *For each segment of the chain, the check’s authority is its decidability: units and diff-negation are decidable and may gate; conservation is noisy under extraction*

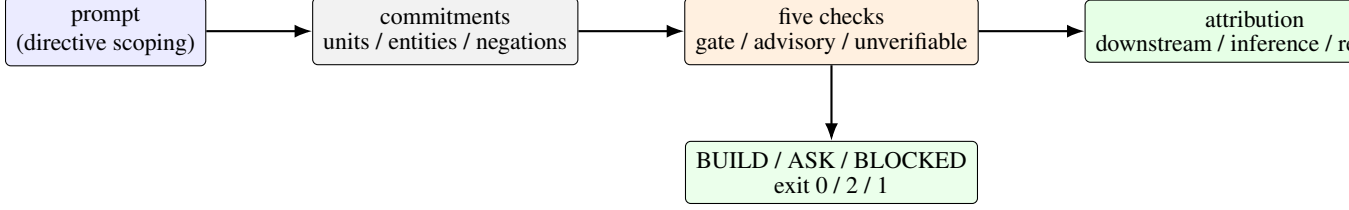


Figure 1: The audit pipeline. Commitments are extracted from directive sentences only; checks emit typed findings; attribution names the broken link; the verdict and exit code are derived, never scored.

and may only advise; whole-response faithfulness is undecidable and may only be instrumented (attribution counts) or declined (unverifiable).

Precedence and exit mapping are fixed and standardized:

verdict	condition	exit code
BUILD	no gate finding; every commitment survived	exit 0
BLOCKED	a gate finding exists (violation)	exit 1
ASK	no gate finding; prompt under-determined	exit 2
USAGE	malformed input or contract violation	exit 3

Precedence: a gate finding makes the verdict BLOCKED *even when* the prompt is also ambiguous—a violation outranks a question (rows read in order; the first satisfied condition wins). Advisory findings never affect the exit code except under an explicit strict flag. A report that says “BUILD with 1 blocking finding” would read as fine next to a failure, so display follows the same precedence.

4 Method

Extraction (all regex, no model). Sentence segmentation, directive scoping, then extraction of numbers-with-units (canonicalized through per-dimension tables: 30 s and 1800 s are the same time), committed entities, and negation targets. Directive scoping is a small formal grammar rather than an open heuristic:

$$\text{sentence} ::= \text{clause} (. \text{clause})^* \quad (1)$$

$$\text{directive} ::= m \text{ clause}, \quad m \in \mathcal{M}, \quad (2)$$

$$\mathcal{M} = \{\text{set}, \text{add}, \text{remove}, \text{must}, \text{ensure}, \text{use}, \text{without}, \text{never}, \dots\} \quad (3)$$

$$\text{commitment} ::= \text{number} [\text{unit}] \mid \text{entity} \mid \text{negation-target}, \quad (4)$$

where a clause is a directive iff it contains a marker from $\mathcal{M}$ in imperative scope, and commitments are extracted only from directives. Multi-clause directives (including conditionals such as “*if retries exceed 3, set timeout to 30s, otherwise use 10s*”) decompose per clause: each clause’s surface commitments are checked for preservation, but *conditional branch selection*—which branch governs at runtime—is outside the modeled segment and is recorded *unverifiable* (or fires ASK when the condition itself is under-determined).

Gate checks (exactly decidable). *Unit adjacency*: if the prompt committed to a quantity with a unit and the response carries the same value with a different unit of that dimension, fire—unit changed: prompt says ‘30 seconds’(s), got min (time). *Unit dropped*: the number survives but the unit vanishes (5 min $\rightarrow$ 5). *Negation added* (diff mode): a forbidden token appears on a + line—the exact “forbidden thing” check. With prose only, a mention under a local decline (*I did not add retries*) is respected; a bare mention is *unverifiable* with the instruction to re-run with a diff, never a failure.

Advisory checks (measured, not timid). *Conservation*: committed entities and bare numbers should survive downstream (committed name lost: ‘parse_config’never appears downstream). Advisory by measurement: on real traffic these fire at 10 $\times$ the gate rate and include paste-and-analyze data; §5 shows the ablation that set the boundary.

Table 1: Boundary measurement on a 113-turn real transcript, and the scoping ablation that set the advisory boundary.

configuration	findings on 113 turns	turns affected
gate checks (shipped)	<code>unit_dropped</code> 22	3%
advisory, every mention anywhere	887 number + 415 entity	50%
advisory, directive scoping (shipped)	415 number + 135 entity	30%
advisory, + copula scoping	780 number + 191 entity	42%

ASK (intervention, not a guess). Four provable under-determinations fire ASK with a question: a committed unitless number (*what unit applies to 4096?*), a vague adjective with no measurable property (*faster measured how?*), a named artifact that does not exist in context, and a negation with no concrete target. The tool refuses to gate on a question it cannot answer—ASK is a question for the person who asked.

Tiering (design discussion, not deployed). On conversational traffic ASK fires on $39/113 = 34.5\%$ of turns; on the task-oriented multi-transcript audit, $163/1,253 = 13.0\%$ (§5)—a rate that would cause alert fatigue if every firing blocked a pipeline. Two levers separate criticality without violating the authority contract: (i) *severity classes*—an unresolvable committed entity or negation target is high-criticality (blocking), while a unitless-number or vague-adjective query can be logged as a warning in asynchronous/CI profiles and surfaced only interactively; (ii) *profile configuration*—blocking vs.log-only, fixed per deployment, never per finding. The shipped contract keeps a single deterministic `exit 2`; tiering is specified here and deferred to future work (§10).

Attribution. Every finding is attributed: gate findings $\rightarrow$ `downstream_of_prompt` (the response violated a determinate prompt); conservation findings $\rightarrow$ `intent_inference` (the extraction scoped wrong); anything else $\rightarrow$ `unattributed`—the measured size of the undecidable slice, the number that says how much risk any gate can reach.

Surfaces. Pair mode (prompt, response, optional diff); transcript mode (`## User / ## Assistant` turns, per-turn findings with turn numbers); multi-file audits; JSON for machines. Identical inputs produce byte-identical reports (tested).

5 Evaluation

Labelled suite (14 cases, both directions). Each check is evaluated against bidirectional labels to ensure both sensitivity and specificity: every check carries must-fire and must-not-fire cases. Score: 13/14, one documented known miss (a copula commitment, discussed below), zero unexpected failures; the six labelled gate cases pass 6/6 and span both directions of all three gate checks (unit adjacency, unit dropped, negation added). The suite is a CI seed suite, not a statistical benchmark: $n = 14$ supports regression gating, not generalization claims. Statistical precision/recall at scale requires a labelled corpus an order of magnitude larger ($n \geq 100$ prompt–response pairs stratified by domain, drawn from public PR diffs); that expansion is the first open challenge (§10). The volume evidence we do have is per-check below: the 1,253-turn audit reports findings per check type, and the suite gates CI—a regression in any check fails the build, not a code review.

The measured boundary. On a 113-turn real transcript (verdicts 74 BUILD / 39 ASK): gates fire on 3/113 turns, advisories on 34/113. Every gate finding was a real `unit_dropped` (prompt says 3s, response carries the bare 3); none were spurious. Table 1 shows the ablation: scoping conservation to directive sentences cut affected turns from 50% to 30%; adding copula commitments (*the budget is 4096*) recovers exactly one labelled case at +89% number findings, so it was measured and rejected. The structural feature that would isolate valid copula declarations is *referent restriction*: a copula complement commits only when its subject head is code-adjacent (snake-case, backticked, path-like, or a known artifact parameter)—narrative copulas whose subjects have no code referent are what produced the +89% noise. That feature is identified here but not shipped; measuring it requires the expanded labelled benchmark. Two scoping attempts is the signal to stop and report: tuning until one transcript is quiet would manufacture unwarranted confidence.

Table 2: Alternatives, measured on our corpora, and the verdict each earned. Labels everywhere come from diffs or held-out splits, never from messages.

alternative	measured	verdict
keyword / regex base-line	$F_1 = 0.090\text{--}0.117$ vs. empirical chance floor 0.1228 (3,000-draw sim., 935 labelled commits); early prototype: 550-finding FP storm	below chance—rejected
bigger encoder	MiniLM 22.7M: 0.3576 ± 0.0321 (5 seeds); 335M: 0.4117; SimCSE worst of three; gain over regex $+0.2504 \pm 0.0384$	capacity buys accuracy, not a new capability
learned commitment classifier	encoder beats keyword control in all 10 runs (mean gain $+0.046/+0.066$) but paired bootstrap $p < 0.05$ in only 3/10; binding constraint: 505 claims, 111 positives	inconclusive—not shipped as a gate

Scale self-audit. 13 transcripts, 1,253 turns: verdicts 1,090 BUILD / 163 ASK; gate findings on 14 turns (1.1%), advisory on 108 (8.6%); findings by check: `number_lost` 248, `entity_lost` 251, `unit_dropped` 24, `unit_adjacency` 1 (a 0.0s $\rightarrow$ h swap). Sample gate rows point at the exact turn (“turn 76: 90 min”), because the contract is that a finding is a pointer to evidence, and review decides.

Contract tests. Fifteen tests pin the exit-code contract (0/1/2/3), diff-enables- negation, strict mode, JSON shape, determinism, and file-input forms. Determinism and the exit contract *are* the API; both are asserted, not documented.

6 Empirical evaluation of alternative approaches

Every obvious alternative was measured before being dismissed (Table 2); each outcome is a measured result.

The standing rule follows directly: a check that cannot clear its own authority bar stays out of the gate set—shipping the classifier anyway would be exactly the authority lie the contract forbids. Real commit messages are indirect (“*Fix #6628 - JSONDecodeError are not deserializable*” $\Leftarrow$ `add_docstring`), which is why lexical matching scores below chance and why the shipped suite parses commitments instead of keywords.

7 Downstream: an exact oracle for training

Because every check is exact, `audit` serves as a labelling oracle for post-training: oracle-derived intent populations (unit swaps, entity loss, negation violations) with contradicted cases *dropped*, *not relabelled*, driving a pre-registered gate (templated intent accuracy $\geq 87\%$) in a companion multi-skill training study [Denadai, 2026] where a 1.5B model moves from 54.5% to 90.9% base-relative intent accuracy. That population includes a dedicated vague-ask family labelled ASK by this engine; the per-family counts (and hence the ASK share of training rows) are pinned in the companion study’s dataset manifest, and because contradicted cases are dropped rather than relabelled, the training share of ASK equals the oracle’s share in the population. The oracle is the reusable artifact; the model is its by-product.

8 Limitations

One corpus, one large transcript for the boundary measurement. Extraction brittleness is real and bounded: directive scoping is English-marker-based, so non-English prompts are entirely outside scope; passive or non-canonical constructions (“*a timeout of 30s should be used*”) fall outside $\mathcal{M}$ ’s imperative scope and yield false negatives; multi-clause conditionals decompose per clause without branch semantics (§4). Dependency parsers or lightweight token labelers would move extraction into the learned tier, where the authority contract—and the labelled-scale requirement—apply unchanged. The copula miss stands as a documented known miss, not a hidden one; prose-only negation is undecidable without a diff and says so; the commitment classifier is inconclusive at current label

counts; the $\sim 35\%$ ASK rate on conversational transcripts is a judgment call that belongs to the requester; and there is no comparison against existing prompt-fidelity tooling—the claim is that the *category* (fidelity vs correctness) is unoccupied where measured, not that no competitor exists anywhere. Extractions surface occasional awkward findings (“12\n S”); a finding is a pointer, and review decides—by design.

9 Conclusion

Commitment fidelity is the question correctness tools skip, and the honest answer has three parts: gate the decidable slice (units, diff-negation—exact, both directions scored, 13/14 with one documented miss), measure the noisy slice (advisory conservation, the 3%/30% boundary set by ablation, not timidity), and instrument the undecidable slice (attribution counts the residual instead of pretending it is zero). The alternatives were measured, not assumed—Table 2 carries their verdicts. The exit code carries the whole design contract—exit 0 through exit 3, never a scalar—ensuring that non-decidable checks emit no misleading confidence and that coverage claims never outrun what the instrument actually decided. Downstream, the payoff is already visible: this oracle’s populations moved a 1.5B model from 54.5% to 90.9% intent accuracy under a pre-registered gate.

10 Future Research and Open Questions

This work was conducted within the hybrid-neurosymbolic research program of **Optimality.ai** (neural proposers, symbolic authorities, fail-closed contracts). The open questions are as follows.

First and binding for statistical claims: expand the labelled benchmark to $n \geq 100$ prompt–response pairs drawn from public corpora (SWE-bench-class PR diffs, GitHub pull requests), stratified by domain, and report per-check precision and recall with intervals—the seed suite ($n = 14$) gates regression, not generalization. Second, measure the referent-restricted copula feature (§5) against that benchmark, and compare the regex pipeline against dependency parsers and lightweight token labelers under the same authority contract. Third, deploy the ASK tiering of §4 (severity classes plus blocking/log-only profiles) and measure interruption rates on task-oriented traffic. Fourth, scale the commitment classifier with diff-derived labels until its bootstrap clears on every split—advisory first, gate authority only when it earns the bar the contract sets. Fifth, grow the transcript corpora so the gate/advisory boundary is measured across domains rather than one fleet; take diff-mode to production traffic so negation checking leaves the undecidable class; and bridge to the proof engine—committed obligations declared as premises the kernel can discharge, closing the loop from “the prompt promised” to “the certificate proves.”

References

- Eduardo Perez Denadai. One model, three oracles: Fail-closed multi-skill post-training from verifiable feedback. In *Optimality.ai Technical Report*, 2026. Companion paper.
- Tianyu Gao, Xingchao Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910, 2021.
- Orla C. Z. Gotel and Anthony C. W. Finkelstein. An analysis of the requirements traceability problem. *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE)*, pages 94–101, 1994.
- Harrison Lee, Hubert Pham, Maggie Lao, et al. RLAIIF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10509–10524, 2023.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krashennnikov, and David Krueger. Defining and characterizing reward hacking. *arXiv preprint arXiv:2212.06305*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36, Datasets and Benchmarks Track*, 2023.

Yongchao Zhou, Mustafa Öztürk, Biao Zhang, et al. Self-reported evaluations: A benchmark for instruction-following. In *arXiv preprint arXiv:2305.11206*, 2023.

Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction scope claims to the decidable slice, the measured boundary, and the documented known miss.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: A dedicated limitations section states single-corpus scope, the copula miss, prose-negation undecidability, inconclusive classifier, and no-fidelity-tooling comparison.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: No theorems beyond the authority-bound proposition (stated with proof); no new formal results are claimed.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Labelled-case suite, both-direction scoring, ablation tables, self-audit counts, and exit-contract tests are fully disclosed; the suite gates CI.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- Reproducibility requires a reasonable avenue for the reported results, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code and transcripts will accompany the final version; the paper discloses the exact checks, scoping rules, labelled cases, and all measured counts.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Sections 3-5 specify extraction, each check’s semantics, precedence, surfaces, and evaluation protocol.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: Both-direction scores on 14 cases, five-seed encoder probe with sd, empirical chance floor via 3,000-draw simulation, paired bootstrap on the classifier, counts with percentages throughout.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).

- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: All checks are regex/stdlib CPU computations over transcripts; the encoder probe is five-seed training of frozen-embedding classifiers on a laptop; costs are negligible and stated.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the code of ethics appropriate to the authors' field and institutions?

Answer: [\[Yes\]](#)

Justification: The authors have reviewed the applicable code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed a code of ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Commitment fidelity catches silent specification drift in AI-generated responses; the honest-unverifiability design limits false confidence.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Regex tooling and small encoders with no high-risk release; findings are pointers for human review.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Encoder checkpoints and the commit corpus are third-party assets used under their published terms and cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The labelled-case suite, transcripts, and corpus are new assets documented for release.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- Under prevailing codes of ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the applicable code of ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Encoders are evaluated components; their role is limited to probes and the paper declares where a model is and is not used.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.