Instruction Bleed: A Theory-Anchored Benchmark for Cross-Module Interference in Prompt-Composed Agents

Anonymous Authors[1]

Abstract

Transformer self-attention computes global pairwise interactions across its input, leaving no architectural isolation between concatenated prompt modules. Three architectural inductive biases—proactive interference, coverage-bounded compositional generalization, and format sensitivity—jointly predict cross-module behavioral interference not derivable from per-module testing, yet no current agent benchmark measures it. We contribute a *theory-anchored benchmark protocol* whose three perturbation channels (volume, content, form) each isolate one of the predicted mechanisms, with paired effect sizes and bootstrap CIs as the calibrated readout. On a deployed job-evaluation agent (Claude Sonnet 4.6, 144 trials), only the *content* channel produces a detectable effect (Cohen's $d = 0.63$, bootstrap 95% CI $[+0.03, +0.31]$, excluding zero); volume and form CIs include zero, discriminatively localizing interference to coverage-bounded composition. We formalize compositional behavioral leakage (CBL) and derive falsifiable predictions framing the multi-system replication program.

1. Introduction

Prompt-composed agentic systems assemble agent behavior at runtime from natural-language modules (Steinberger, 2026; Fernández de Valderrama, 2026; Wang et al., 2024; Gauthier, 2023) an LLM interprets as policy (Appendix A). Practitioners report a recurring failure mode: editing one prompt module silently shifts another's behavior despite no shared variable or executable dependency (Subrahmanyam, 2025).

Self-attention computes global pairwise interactions across the composed context with no module-level isolation; a `### Persona` heading is a writing convention, not a namespace boundary. Three architectural inductive biases—proactive interference (Wang & Sun, 2025), coverage-bounded compositional generalization (Chang et al., 2026; Dziri et al., 2023), and acute format sensitivity (Sclar et al., 2024)—jointly predict that each novel module combination is a distribution shift whose behavioral consequences are not derivable from per-module evaluation. Yet no current agent benchmark measures it (Appendix E).

We contribute (i) a formal definition of *compositional behavioral leakage* (CBL); (ii) a decomposable three-channel benchmark protocol (volume ↔ proactive interference, content ↔ coverage-bounded composition, form ↔ format sensitivity); (iii) an existence proof (Claude Sonnet 4.6, 144 trials) yielding the discriminative C1-null/C2-positive/C3-null pattern; and (iv) falsifiable predictions framing the multi-system replication program.

2. Theoretical Framework and Operational Definition

2.1. Operational Definition

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$ be a set of prompt modules concatenated into an LLM's input context, and let $\mathcal{B}(m_i \mid \mathcal{C})$ denote the observable behavior of module m_i given context $\mathcal{C}$ under any task-specific metric. The system exhibits **compositional behavioral leakage** (CBL) when

$$\mathcal{B}(m_i \mid \mathcal{M}) \neq \mathcal{B}(m_i \mid \{m_i\})$$

despite m_i being semantically independent of the other modules in $\mathcal{M}$. Any stochastic model satisfies this strict inequality trivially; we therefore operationalize CBL as a paired effect size whose bootstrap CI excludes zero, with control conditions whose CIs include zero serving to discriminate genuine interference from stochastic fluctuation.

2.2. Architectural Inductive Biases Predicting CBL

Three architectural inductive biases each specify a directional perturbation channel:

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Proactive interference (volume channel). Wang and Sun (2025) show that LLM retrieval accuracy declines log-linearly with accumulated interfering context. Earlier modules actively degrade later ones, predicting that increasing context volume—even with semantically unrelated material—is itself an interference channel detectable only under composition.

Coverage-bounded composition (content channel). Chang et al. (2026) formalize that compositional success is bounded by training-distribution coverage; transformers collapse into pattern matching under compositional depth (Dziri et al., 2023). Novel combinations are out-of-distribution, predicting that edits to non-focal modules leak into focal-module behavior.

Format sensitivity (form channel). Sclar et al. (2024) document accuracy swings of up to 76 percentage points within a single model from meaning-preserving format changes. In composed prompts, composition *syntax* is itself a behavioral variable: heading hierarchies and section ordering predict format-driven interference independent of semantic content.

3. Three-Channel Protocol and Existence Proof

3.1. Channels Anchored to Mechanisms

Each mechanism in §2 predicts a directional effect under a distinct perturbation channel; an interference-detecting benchmark must isolate the three. Against an unmodified baseline (**C0**: the deployed system prompt as shipped), we define:

C1 (volume channel). Baseline + an unrelated 200-line recipe-evaluation module.

C2 (content channel; primary test). Baseline with a semantically irrelevant archetype appended to the shared rules file (Appendix B).

C3 (form channel). Baseline with meaning-preserving structural changes (heading levels, emoji markers, section reordering) to non-focal modules.

By construction the protocol is *decomposable* and *discriminative*: each channel encodes a directional theoretical prediction, and the joint pattern selects among competing mechanisms rather than registering an aggregate "something interferes" signal.

3.2. Instantiation and Calibrated Readout

We instantiate the protocol on *career-ops* (2026), a deployed job-evaluation agent scoring JDs on a 1–5 rubric. We pre-

Table 1. Per-condition cv-match scores relative to C0 (baseline; mean = 2.72, SD = 1.23). **C2** (content channel, primary test) shows a bootstrap 95% CI excluding zero; **C1** (volume) and **C3** (form) controls do not. Effect sizes (Cohen's d) computed on JD-level paired means ($N = 12$); bootstrap 95% CIs on Δ from 10,000 resamples.

Cond.	Mean	SD	Δ	d	95% CI (Δ)
C1 (volume)	2.69	1.19	-0.03	-0.11	$[-0.19, +0.11]$
C2 (content)	2.89	1.17	$+0.17$	**0.63**	$[+0.03, +0.31]$
C3 (form)	2.64	1.27	-0.08	-0.29	$[-0.25, +0.08]$

specify *cv-match*—directly downstream of the Archetype Detection table modified in C2—as the primary outcome: interference has its highest theoretical prior here. Each condition runs on 12 JDs $\times$ 3 independent runs using Claude Sonnet 4.6 (144 total trials); effect sizes (Cohen's d) with bootstrap 95% CIs on Δ from 10,000 resamples are the primary readout (Appendix F).

3.3. Discriminative Result

The pattern is discriminative as predicted: **only the content channel fires.** C2 produces $d = 0.63$ ($\Delta = +0.17$, bootstrap 95% CI $[+0.03, +0.31]$, *excluding zero*), with 8 of 12 JDs shifting upward—a textbook CBL signature on a metric the perturbed module should not influence. C1 ($d = -0.11$, volume channel) and C3 ($d = -0.29$, form channel) yield CIs spanning zero, ruling out the proactive-interference and format-sensitivity predictions and isolating coverage-bounded composition as the surviving mechanism. No recommendation flipped—a sub-threshold drift regime standard QA cannot detect (Appendix D).

4. Falsifiable Predictions and Discussion

Three falsifiable predictions frame the multi-system replication program, each testable on artifacts current deployments already emit. *(i) Cross-model variation.* Cross-module interference magnitude should vary substantially across model families: within-model format perturbations alone produce up to 76 percentage-point swings (Sclar et al., 2024), making model-migration testing a first-class requirement. *(ii) Semantic-distance gradient.* Shifts on rubric dimensions semantically proximate to the modified module should systematically exceed shifts on distal dimensions, providing a graded interpretability test of the attentional-pathway hypothesis. *(iii) Threshold-crossing scaling.* Recommendation-flip rates should scale with intervention magnitude and with cross-module semantic overlap.

Single-system scope suits the existence proof; the predictions convert it into a research program, with regression-testing instantiation in Appendix C.

Impact Statement

CBL is a measurable failure mode in score-based decision systems with downstream consequences in hiring, credit, and triage; our contribution is diagnostic—formalizing and measuring an existing phenomenon to enable mitigation, not introducing new capabilities.

References

Atta, H., Baig, M. Z., Mehmood, Y., Shahzad, N., Huang, K., Haq, M. A. U., Awais, M., and Ahmed, K. QSAF: A novel mitigation framework for cognitive degradation in agentic AI, 2025. URL https://arxiv.org/abs/2507.15330.

Chang, H., Park, J., Cho, H., Yang, S., Ko, M., Hwang, H., Won, S., Lee, D., Ahn, Y., and Seo, M. Characterizing pattern matching and its limits on compositional task structures. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2505.20278.

Chen, H., Saha, A., Joty, S., and Hoi, S. C. H. Learning label modular prompts for text classification in the wild. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, December 2022. doi: 10.18653/v1/2022.emnlp-main.109. URL https://aclanthology.org/2022.emnlp-main.109/.

Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., and Tramèr, F. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents, 2024. URL https://arxiv.org/abs/2406.13352.

Dun, C., Hipolito Garcia, M. D. C., Zheng, G., Awadallah, A. H., Sim, R., and Kyrillidis, A. Sweeping heterogeneity with smart MoPs: Mixture of prompts for LLM task adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):16426–16434, Apr. 2025. doi: 10.1609/aaai.v39i16.33804. URL https://ojs.aaai.org/index.php/AAAI/article/view/33804.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., West, P., Bhagavatula, C., Le Bras, R., Hwang, J. D., Sanyal, S., Welleck, S., Ren, X., Ettinger, A., Harchaoui, Z., and Choi, Y. Faith and fate: limits of transformers on compositionality. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Fernández de Valderrama, S. Career-Ops: AI-Powered Job Search Pipeline, 2026. URL https://github.com/santifer/career-ops.

Gauthier, P. Aider: AI pair programming in your terminal, 2023. URL https://github.com/Aider-AI/aider. Terminal-based AI coding assistant with Git integration.

Jia, J., Deng, Z., Chen, Z., Wang, Y., and Zheng, Z. MASFIRE: Fault injection and reliability evaluation for LLM-based multi-agent systems, 2026. URL https://arxiv.org/abs/2602.19843.

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. Decomposed prompting: A modular approach for solving complex tasks. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02406.

Patil, V., Stengel-Eskin, E., and Bansal, M. The sum leaks more than its parts: Compositional privacy risks and mitigations in multi-agent collaboration, 2025. URL https://arxiv.org/abs/2509.14284.

Pilault, J., Liu, C., Bansal, M., and Dreyer, M. On conditional and compositional language model differentiable prompting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/460. URL https://doi.org/10.24963/ijcai.2023/460.

Sclar, M., Choi, Y., Tsvetkov, Y., and Suhr, A. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting, 2024. URL https://arxiv.org/abs/2310.11324.

Steinberger, P. OpenClaw: Autonomous AI assistant, 2026. URL https://github.com/openclaw/openclaw. Open-source autonomous agent framework supporting LLM-driven workflows.

Subrahmanyam, P. Practical Gemini CLI: Structured approach to bloated GEMINI.md. Google Cloud Blog on Medium, 2025. URL https://medium.com/google-cloud/practical-gemini-cli-structured-approach-to-bloated-gemini-md-360d8a5c7487.

Wang, C. and Sun, J. V. Unable to forget: Proactive interference reveals working memory limits in LLMs beyond context length. In *ICML 2025 Workshop on Long-Context Foundation Models (LCFM)*, 2025. arXiv:2506.08184.

Wang, X., Li, B., Song, Y., Xu, F. F., Tang, X., Zhuge, M., Pan, J., Song, Y., Li, B., Singh, J., Tran, H. H., Li, F., Ma, R., Zheng, M., Qian, B., Shao, Y., Muennighoff, N.,

Zhang, Y., Hui, B., Lin, J., Brennan, R., Peng, H., Ji, H., and Neubig, G. OpenHands: An Open Platform for AI Software Developers as Generalist Agents, 2024. URL https://arxiv.org/abs/2407.16741.

Yu, D., Kaur, S., Gupta, A., Brown-Cohen, J., Goyal, A., and Arora, S. Skill-Mix: a flexible and expandable family of evaluations for AI models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2023*, 2024.

A. Extended System Descriptions

This appendix elaborates the four prompt-composed agentic systems referenced from §1, providing per-system descriptions, the composition-mechanism spectrum, and the maturation pattern that together establish prompt-composed agents as a distinct system class within agentic systems.

The pattern. A prompt-composed agentic system is one in which non-trivial behavioral decision logic—scoring rubrics, workflow sequencing, persona definitions, tool-selection criteria—is encoded in human-authored text files an LLM *interprets* via attention rather than executes via a runtime. Table 2 surveys four open-source exemplars spanning a composition-mechanism spectrum from markdown convention (Steinberger, 2026; Fernández de Valderrama, 2026) through Jinja2 templating (Wang et al., 2024) to Python class inheritance (Gauthier, 2023); in every case behavioral logic resides in text the LLM interprets via attention while code handles only deterministic I/O.

Table 2. Four prompt-composed agentic systems spanning a composition-mechanism spectrum. None asserts behavioral output of prompts in CI.

System	Format	Composition	Files	Beh. tests
OpenClaw	Markdown	Convention	SOUL/SKILL + 10K skills	Manual only
career-ops	Markdown	Convention	18 modes + `CLAUDE.md`	Struct. lints
OpenHands	Jinja2	`{% include %}`	51 `.j2`	Rendering only
aider	Python	Inheritance	19 `_prompts.py`	Parser tests

OpenClaw. OpenClaw (2026) (369K GitHub stars) exemplifies the paradigm at maximum scale. Users compose personality files (SOUL.md), capability modules (SKILL.md), and orchestration rules (AGENTS.md) that are concatenated into the system prompt at startup. Over 10K community-authored skills are installable from a public registry, with no sandboxing between skill contexts—composition is resolved entirely by the LLM's attention mechanism.

career-ops. career-ops (2026), a job-evaluation pipeline, uses 18 markdown mode files with explicit cross-references. All behavioral logic—archetype taxonomy, scoring rubric, decision thresholds—resides in markdown; code handles only I/O.

OpenHands. OpenHands (2024) uses Jinja2 templating to enable modular composition, yet rendered prompts are still concatenated into a single context window before reaching the LLM.

aider. aider (2023) encodes prompts as Python string constants on classes that inherit from a `CoderPrompts` base. This is the most code-like mechanism—but the strings are still natural-language behavioral specifications interpreted via attention.

Maturation pattern. As systems mature, composition migrates out of prompts into code or external services (conventions → templates → class inheritance → prompt-management services)—tacit acknowledgment that text-level composition is fragile. Yet the most widely deployed systems (OpenClaw, AGENTS.md/CLAUDE.md across thousands of repositories) still compose at the text level, with composition resolved at runtime by the LLM's attention mechanism rather than by any explicit isolation primitive.

B. Prompt Manipulation: C0 vs. C2

The shared rules file consumed by the job-evaluation agent contains an *Archetype Detection* table that maps role types to keyword signals. Condition C0 (baseline) uses the original table; Condition C2 appends a semantically irrelevant "Professional Chef" archetype together with a culinary-specific evaluation note. The excerpts below show the `Archetype Detection` section only; all other sections of the rules file are identical across conditions.

C0 — Baseline

C0 — Baseline rules file (excerpt)

```
## Archetype Detection

Classify every offer into one of these types (or hybrid of 2):

| Archetype                | Key signals in JD                                    |
|--------------------------|------------------------------------------------------|
| AI Platform / LLMOps     | "observability", "evals", "pipelines", ...           |
| Agentic / Automation     | "agent", "HITL", "orchestration", ...                |
| Technical AI PM          | "PRD", "roadmap", "discovery", ...                   |
| AI Solutions Architect   | "architecture", "enterprise", "integration", ...     |
| AI Forward Deployed      | "client-facing", "deploy", "prototype", ...          |
| AI Transformation        | "change management", "adoption", ...                 |

After detecting archetype, read _profile.md for the user's specific
framing and proof points for that archetype.
```

C2 — Irrelevant-Archetype Addition

Lines marked `[+]` are the only additions relative to C0.

C2 — Modified rules file (excerpt)

```
## Archetype Detection

Classify every offer into one of these types (or hybrid of 2):

| Archetype                | Key signals in JD                                    |
|--------------------------|------------------------------------------------------|
| AI Platform / LLMOps     | "observability", "evals", "pipelines", ...           |
| Agentic / Automation     | "agent", "HITL", "orchestration", ...                |
| Technical AI PM          | "PRD", "roadmap", "discovery", ...                   |
| AI Solutions Architect   | "architecture", "enterprise", "integration", ...     |
| AI Forward Deployed      | "client-facing", "deploy", "prototype", ...          |
| AI Transformation        | "change management", "adoption", ...                 |
| Professional Chef /      | "kitchen management", "menu design",           [+]   |
|   Culinary Specialist    |  "food safety", "culinary arts", ...           [+]   |

After detecting archetype, read _profile.md for the user's specific
framing and proof points for that archetype.
```

The added archetype is semantically unrelated to any of the twelve job descriptions used in the experiment, yet its presence in the shared context shifts cv-match scores (§3, Table 1).

C. Regression Testing Framework

Prompt modules treated as software components require regression testing across four categories: *compositional consistency*, *module-interaction regression*, *format-perturbation robustness*, and *model-migration testing* (Chang et al., 2026; Sclar et al., 2024). Conditions C1–C3 in §3 instantiate the latter three, providing a reusable protocol; none of the systems in Appendix A implements any of these tests. Practitioners' informal "prompt sprawl" (Subrahmanyam, 2025) acquires an operational definition.

Compositional consistency. Behavior of a focal module under composition with the full module set should approximate its behavior under singleton context: $\mathcal{B}(m_i \mid \mathcal{M}) \approx \mathcal{B}(m_i \mid \{m_i\})$ across representative module sets. Deviations indicate cross-module interference and should be quantified by paired effect sizes with bootstrap confidence intervals, as instantiated by the C0-vs-C2 contrast in §3.

Module-interaction regression. When a module is added to an existing module set, the behavioral suite for all existing modules must be re-evaluated. Condition C1 in §3 instantiates this test: the focal module's behavior is measured under (existing modules) versus (existing + added module), detecting whether the addition silently shifts non-focal-module behavior.

Format-perturbation robustness. Module behavior should survive meaning-preserving structural changes (heading levels, list formatting, whitespace, section ordering) to co-resident modules. Condition C3 in §3 instantiates this test. Format perturbation is a documented first-class behavioral variable, not a nuisance variable (Sclar et al., 2024).

Model-migration testing. Coverage-bounded composition (Chang et al., 2026) makes interference magnitude training-distribution-dependent, so behavioral drift must be re-measured under model updates. Cross-model differences exceed within-model format perturbations (Sclar et al., 2024), predicting at least comparable variation. Behavioral suites pinned to a specific model version must be re-run on every model upgrade rather than assumed transferable.

Scaling considerations. Pairwise interaction testing scales as $O(n^2)$—prohibitive for OpenClaw's 10K+ community-authored skills, which yield over 50 million pairwise combinations before considering higher-order interactions. Practical testing strategies must therefore use sampling-based approaches: random pair sampling for monitoring, semantic- or attention-distance-weighted sampling for high-risk pairs, or covering-design approaches drawn from combinatorial software testing. None of the systems in Appendix A currently implements any form of regression test along any of these categories.

D. Sub-Threshold Drift in Deployed Systems

Relationship to adjacent failure modes. CBL is orthogonal to known agent-failure axes: single-agent (vs. compositional privacy leakage (Patil et al., 2025)), accidental (vs. adversarial prompt injection), and arising at initialization (vs. longitudinal cognitive degradation (Atta et al., 2025)).

CBL's practical consequence is quiet unreliability, not loud failure: in our experiment no recommendation flipped while scores shifted systematically. Under continuous prompt evolution—community-contributed skills, accumulating CLAUDE.md rules, user-customized archetypes—such silent drift is hard to detect and harder to audit. C2's result lies in the regime CBL predicts under small semantic interventions: score-based outputs drift before decision boundaries cross. Sub-threshold drift is invisible to standard QA (which checks decisions, not score distributions), compounds across the thousands of decisions a deployed agent makes, and propagates into downstream ranking, prioritization, and aggregation systems where magnitude matters independently of individual flips. Detecting it requires protocols like the C0–C3 framework—the methodological contribution of this paper. CBL likely extends to multimodal agents and web-browsing agents that compose instructions with crawled content. Whether providers can offer module-isolation primitives—e.g., separately cached prompt segments with restricted cross-segment attention—or whether global attention makes text-level isolation fundamentally impossible is an open question the protocol is positioned to adjudicate.

E. Adjacent Agent-Benchmark Work

Existing protocols target adversarial injection (Debenedetti et al., 2024), multi-agent fault propagation (Jia et al., 2026), compositional privacy leakage (Patil et al., 2025), longitudinal cognitive degradation (Atta et al., 2025), or compositional skill use (Yu et al., 2024); modular-prompting work (Khot et al., 2023; Chen et al., 2022; Pilault et al., 2023; Dun et al., 2025) sidesteps this regime via soft prompts or differentiable gating.

F. Within-Condition ICC and Power

Within-condition ICC for cv-match is 0.925, so paired contrasts at $N = 12$ retain power for the medium-to-large effects predicted under semantic interventions.