Prompt Drift Lab: An Auditable Evaluation of Structured Instruction Following
Under Benign Prompt Variants and Instruction Explicitness

YuChen Zhu
Shanghai University

Abstract

Single-prompt evaluation is often treated as a stable proxy for instruction-following ability and structured-output reliability. We present **Prompt Drift Lab**, an audit-style evaluation that holds tasks and decoding settings fixed while applying *benign* prompt perturbations and an *implicit vs. explicit* instruction condition.

We evaluate three generation models (ChatGPT, Claude, Gemini) on two questions (Q3–Q4) with four Prompt-B variants (`baseline`, `long`, `weak`, `conflict`) and two instruction triggers (`explicit`, `implicit`), yielding 16 outputs per model. Each output is A/B-blind and scored by two out-of-family judges (cross-model judging), with a self-evaluation supporting check.

Harmless prompt changes can shift conclusions by multiple points: for ChatGPT, mean total score moves from 6.125 (`baseline`) to 9.625 (`conflict`) (+3.500), while Gemini drops from 5.375 to 4.750 (-0.625). Instruction explicitness is an even stronger lever: averaged across variants, Gemini scores 9.3125 under `explicit` triggers but only 0.500 under `implicit` triggers (gap 8.8125); Claude scores 4.375 vs. 0.000. We do not propose a universal fix; instead, we release an auditable protocol and artifacts showing that single-prompt conclusions can be overconfident under benign rewordings and constraint phrasing.

1 Introduction

Many evaluations of large language models rely on a single "canonical" prompt. This practice implicitly assumes that small, semantically equivalent edits (rewording, reordering, strengthening/weakened phrasing) should not materially change evaluation outcomes. In reality, practitioners frequently report *format breaks*, *instruction omissions*, and *semantic drift* after minor prompt edits.

This paper takes an **audit stance**. We do not claim causal mechanisms for drift, nor do we claim broad generalization across all tasks or models. Instead we ask:

> **Audit question.** How stable are structured instruction-following evaluation results under benign prompt perturbations and instruction explicitness changes, when tasks and decoding settings are held fixed?

We contribute: (1) a minimal, reproducible evaluation suite with versioned tasks and prompt variants, (2) a five-dimensional rubric capturing structure and actionability, and (3) fully auditable run artifacts enabling re-computation of every reported number.

Claim boundary. Our claims are limited to the included tasks (Q3–Q4), prompt family B and its variants, the evaluated models, and the documented protocol. We do not infer underlying causes or extrapolate to other settings.

2 Setup

2.1 Tasks and Conditions

We evaluate two questions (Q3–Q4) focused on instruction-following robustness under format constraints and prompt-length folk claims. For prompt stimuli, we use **Prompt Family B** (protocol-ready three-section output contract) and apply four variants: `baseline` (reference), `long` (longer/redundant constraints), `weak` (weakened constraints), `conflict` (tension-inducing instructions).

We also introduce an **instruction explicitness trigger**: `explicit` makes the three-section structure mandatory; `implicit` suggests/implies the structure. This yields 2 questions ×4 variants ×2 triggers $= 16$ outputs per model.

2.2 Rubric (A–E) and Scoring

Each output is scored on five dimensions (0–2 each, total 0–10): (A) **Structure** (three sections present and ordered), (B) **Snapshot constraint** (fact snapshot includes required constraints), (C) **Actionability** (ChatGPT web-search instruction is executable), (D) **Completeness** (Gemini deep-research instruction is complete), (E) **Drift control** (no out-of-protocol behavior).

2.3 Judging Protocol and Auditability

We report **Main method (cross-model judging)**: each generator output is independently scored by two other models (A/B-blind; no cross-sample comparison). We also run a **Supporting method (self-eval)** as a sanity check. All artifacts are stored in a fixed layout (raw PDFs, judged JSON, and aggregated CSV tables) enabling end-to-end audit.

3 Results

3.1 R1: Overall performance differs sharply under implicit vs. explicit triggers

Table 1 summarizes main-method scores. Explicit triggers are consistently high for ChatGPT and Gemini, while implicit triggers collapse for Claude and often for Gemini.

Generator	Mean total	Explicit avg	Implicit avg	n outputs
ChatGPT	8.5625	9.3750	7.7500	16
Claude	2.1875	4.3750	0.0000	16
Gemini	4.9062	9.3125	0.5000	16

Table 1: Main method (cross-model judging): aggregated totals (0–10) and explicit/implicit breakdowns.

Figure 1 visualizes the explicit–implicit gap.

3.2 R2: Benign prompt variants can shift conclusions by multiple points

Prompt variants produce substantial score shifts (Table 2). Notably, ChatGPT moves from 6.125 (`baseline`) to 9.625 (`conflict`), while Gemini drops from 5.375 to 4.750 across non-baseline variants.

3.3 R3: Cross-judge agreement is non-trivial (and should be reported)

Because evaluation itself can drift, we measure inter-judge agreement on total scores (Table 3). ChatGPT and Gemini judges agree closely on this suite (MAE 0.125), while other judge pairs differ more.

3.4 R4: Supporting self-evaluation shows similar qualitative patterns

The supporting method (self-evaluation) provides a sanity check that the instability is not solely an artifact of cross-judge pairing. Supporting summaries are released alongside the main tables.

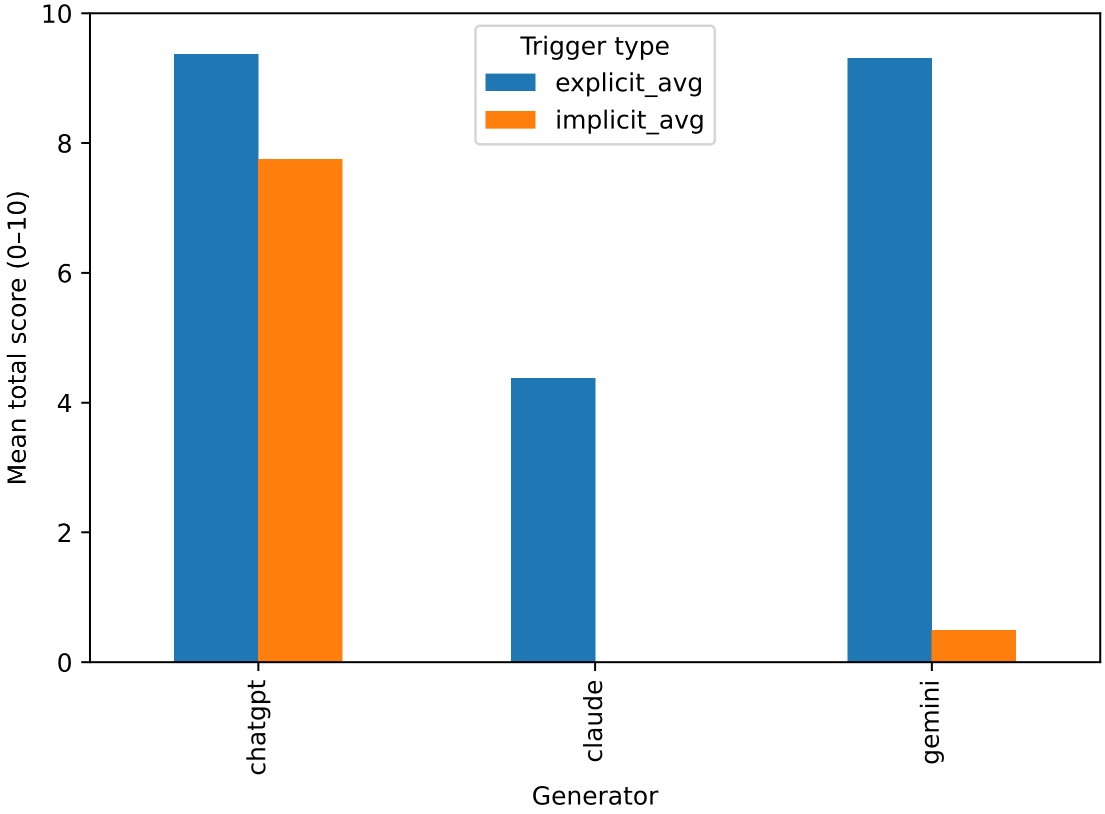

Figure 1: Mean total score (0–10) under explicit vs. implicit triggers (main method).

4 Discussion

Implication for evaluation practice. Our results show that single-prompt evaluations can be *overconfident*: benign prompt variants and the explicitness of constraints can change structural compliance and actionability dramatically. Reporting only a single score can hide whether failures are due to format collapse, missing constraints, or drift outside the intended protocol.

Protocol recommendations (non-binding). At minimum, we recommend: (1) evaluating a small *prompt set* rather than a single prompt, (2) reporting explicit vs. implicit conditions when format contracts matter, (3) including inter-judge agreement or judge-sensitivity analysis, and (4) releasing auditable artifacts.

5 Limitations

This audit is intentionally small (two questions, one prompt family, three models). We do not claim generalization beyond the documented scope. If LLMs are used as judges, their behavior can also drift; we therefore report agreement statistics and release raw evidence for independent auditing.

6 Conclusion

Prompt Drift Lab provides a reproducible, artifact-driven audit showing that structured instruction-following evaluations are not stable under benign prompt drift and instruction explicitness changes. The contribution is protocol-level: we provide a minimal suite, rubric, and evidence chain to help researchers avoid misleading single-prompt conclusions.

Generator	Variant	Mean total	Δ vs. baseline
ChatGPT	baseline	6.125	0.000
ChatGPT	conflict	9.625	+3.500
ChatGPT	long	9.375	+3.250
ChatGPT	weak	9.125	+3.000
Claude	baseline	2.125	0.000
Claude	conflict	2.250	+0.125
Claude	long	2.125	+0.000
Claude	weak	2.250	+0.125
Gemini	baseline	5.375	0.000
Gemini	conflict	4.750	-0.625
Gemini	long	4.750	-0.625
Gemini	weak	4.750	-0.625

Table 2: Main method: mean total score by prompt variant (averaged over Q3–Q4 and triggers).

Judge A	Judge B	n	MAE(total)	Exact match
ChatGPT	Claude	16	1.0625	7/16
ChatGPT	Gemini	16	0.1250	14/16
Claude	Gemini	16	1.2500	6/16

Table 3: Main method: inter-judge agreement on total score.

Reproducibility

All numbers in this paper are computed from released CSV summaries derived from judged outputs. The repository includes: versioned prompt assets, evaluation rules, raw model outputs (PDF), judge JSON, and summary tables.

References

A Directory Layout (Auditable Artifacts)

```
01_experiment_design/   (tasks, protocol)
02_prompt_variants/     (prompt family + variants)
03_evaluation_rules/    (rubric + judge I/O schema)
04_results/
  01_raw_model_outputs/ (PDF outputs)
  02_cross_model_evaluation/
    valid_evaluations/  (judge JSON + summary tables)
    invalid_evaluations/
```

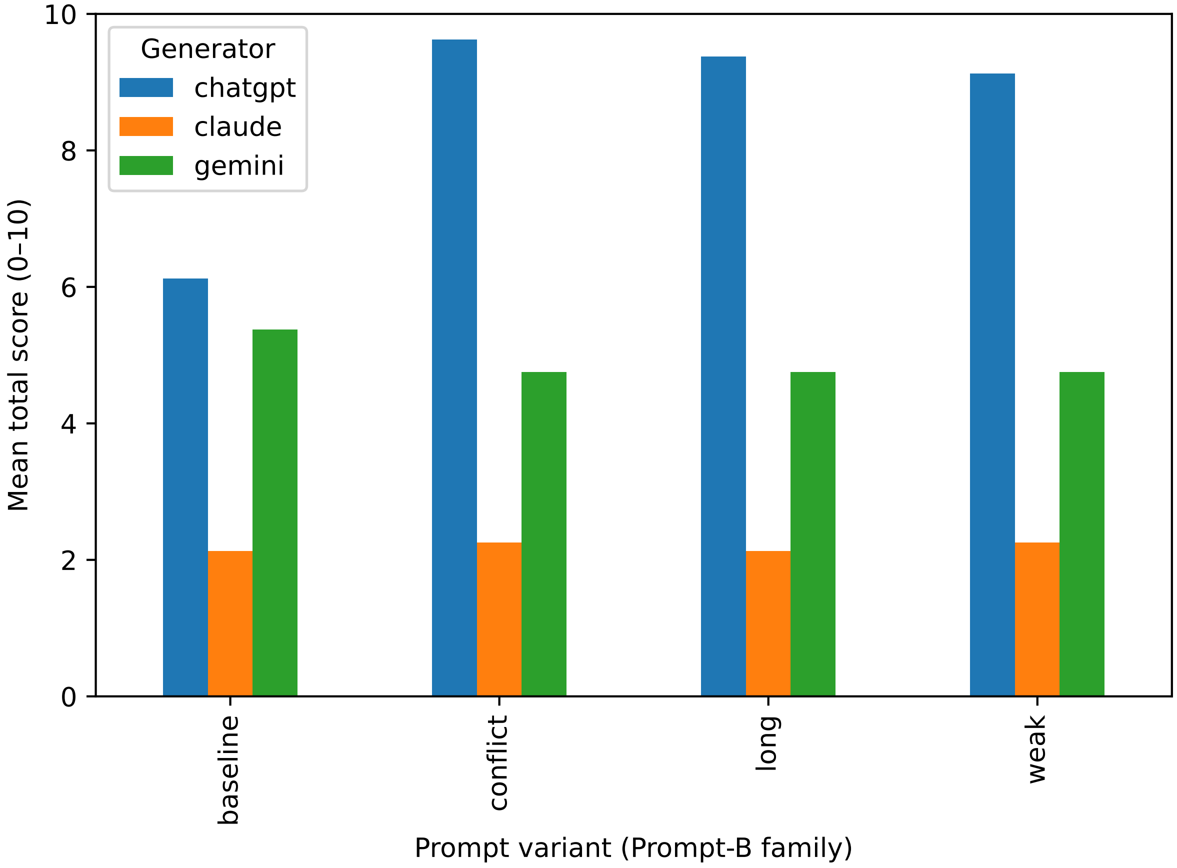

Figure 2: Mean total score (0–10) by prompt variant for each generator (main method).