

Source-Preserving Prompt Augmentation and Category-Adaptive Inference for Qwen3-1.7B

A Prospective Seed-Replication Study

Thibaud Peverelli

Independent Researcher — Artificial Intelligence Systems

United States

Correspondence: tpeverelli@axali-consulting.com

Preprint · Version 1.1 · 29 August 2026

DOI: <https://doi.org/10.5281/zenodo.22081441>

Abstract

Can a fixed compound intervention improve a small language model without updating its parameters? We report a sequential experimental program that separates structured prompt replacement, natural-language rewriting, source-preserving semantic augmentation, and a category-adaptive Qwen inference profile. Public evaluation used 48 IFBench and 48 LiveBench tasks with research-authored semantic-stress variants, pinned Qwen3 revisions, programmatic scorers, matched stochastic seeds, and raw evidence frozen before analysis. Structured replacement was harmful in P1.1, and Qwen-oriented rewriting was null in P1.2. P1.3 then prospectively tested the previously exploratory package-level comparison on the same 96 semantic-stress tasks and frozen augmentations, but with two new seed roots. Qwen3-1.7B improved from 0.272 under raw non-thinking inference to 0.360 under source-preserving augmentation plus the category-adaptive inference profile, a paired gain of 0.088 (95% percentile-bootstrap CI 0.018 to 0.160; 24 wins, 63 ties, 9 losses). The result met the prespecified positive-classification rule and remained positive in complete-case and contract-aware sensitivity analyses. The confidence interval excluded zero but did not establish that the population effect exceeds the prespecified 0.05 practical threshold. Component analysis showed that the adaptive inference profile alone contributed +0.061 (95% CI 0.021 to 0.107), whereas augmentation under matched adaptive inference contributed +0.027 (95% CI -0.046 to 0.099); the interaction estimate was -0.018. Thus, the experiment confirms the complete intervention on this fixed task set and model, not the independent causal contribution of semantic augmentation. The package increased prompt length, latency, and test-time computation and did not establish a quality-cost Pareto improvement. The findings support source-preserving compound inference as a promising mechanism for small models while leaving external generalization and component attribution open.

Keywords: compound AI systems; source-preserving augmentation; adaptive inference; test-time compute; small language models; Qwen3; prospective seed replication

1. Introduction

Increasing model scale remains a reliable way to improve language-model capability, but it also raises memory requirements, serving cost, hardware concentration, and energy use. A complementary strategy is to improve the system around the model: requests can be interpreted, routed, augmented, verified, or assigned additional inference-time computation before an answer is produced. This systems perspective motivates compound AI architectures in which observed capability depends not only on parameters, but also on representation and control [2-4,16,17].

Prompt transformation is an obvious control mechanism, yet a representation that appears cleaner to a human can alter information salience, remove useful redundancy, or move the task away from the downstream model's learned input distribution. A safer alternative may be to preserve the task-bearing source and add a compact, source-supported control layer that highlights active constraints and precedence without reconstructing the entire problem.

We study that distinction through a sequential program. The initial pilot exposed ceiling effects and evaluator defects. P1.1 then tested structured replacement across Qwen3-1.7B, 4B, and 8B and found that replacing the source with a TaskPacket rendering was harmful. P1.2 tested Qwen-native rewriting, source-preserving semantic augmentation, and a category-adaptive inference profile. Rewriting alone was null, while the complete augmented/adaptive package produced an exploratory gain of 0.112 objective-score points for Qwen3-1.7B. P1.3 prospectively specified that package-level comparison as its primary endpoint and evaluated the same fixed tasks and augmentations under two new sampling seeds.

P1.3 met its prespecified positive-classification rule. Source-preserving guidance plus the original request and the category-adaptive inference profile improved Qwen3-1.7B by 0.088 absolute objective-score points, a 32% increase relative to the raw non-thinking mean. Both seed estimates were positive, and complete-case and contract-aware analyses preserved the conclusion. The design is a prospective seed replication: it tests robustness to new stochastic draws, not generalization to new tasks, compiler outputs, model families, or independent implementations.

The study makes four contributions. First, it shows that structured semantic state is not automatically a suitable replacement interface for an answer model. Second, it formalizes a control-plane/data-plane architecture in which the original request remains authoritative. Third, it prospectively confirms a package-level gain from source-preserving augmentation combined with a category-adaptive Qwen inference profile on a fixed Qwen3-1.7B setting. Fourth, it decomposes the package and shows that the adaptive profile accounts for a substantial share of the mean gain, while the incremental augmentation effect under matched inference remains uncertain.

2. Related Work

2.1 Prompt optimization and compilation

OPRO uses a language model as an optimizer that proposes instructions from observed task scores [6], while DSPy compiles declarative language-model programs into prompt and demonstration configurations [5]. Self-Refine improves outputs through iterative feedback [14]. These approaches show that model behavior can be changed without updating weights, but they typically optimize a recurring task or pipeline. The present work instead compiles individual user requests and separates the compiler's internal semantic state from the downstream model's final input representation.

Prompt optimization is also fragile. Small-model replications of OPRO report limited optimizer capability [7], and PromptRobust demonstrates that modest prompt perturbations can substantially change performance [8]. These findings motivate paired experiments in which model revision, task, and stochastic seed are controlled while only representation and reasoning policy vary.

2.2 Compound systems, routing, and model selection

Compound AI systems combine models with planners, retrievers, tools, and execution policies [2]. FrugalGPT and RouteLLM allocate requests across models to trade accuracy against cost [3,4], while Mixture-of-Agents aggregates multiple model outputs [16]. The present study examines a lower-level control problem: before choosing or invoking an answer model, can a semantic front end improve the information and reasoning policy delivered to it?

2.3 Test-time reasoning and adaptive compute

Chain-of-thought prompting established that intermediate reasoning can improve difficult tasks [12]. Test-time-compute studies and reasoning-teacher approaches further treat reasoning effort as a separable systems variable rather than a fixed property of model weights [13,15]. Qwen3 exposes thinking and non-thinking modes within one model family [1,19], allowing category-based allocation of an inference profile. In our experiments, the adaptive profile was fixed before acquisition and did not depend on observed answers.

2.4 Objective instruction following and live evaluation

IFEval introduced programmatically verifiable instruction-following constraints [9], and IFBench extends this paradigm to out-of-domain constraints [10]. LiveBench provides objectively scored and periodically refreshed tasks spanning coding, reasoning, mathematics, data analysis, instruction following, and language [11]. These benchmark families reduce dependence on subjective model judging and support paired representation experiments.

3. Compound Intervention and Causal Scope

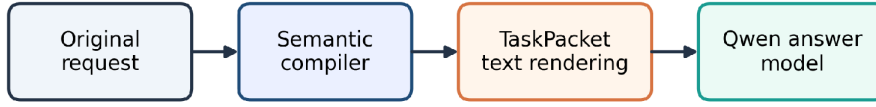
Let x denote the original request, C a semantic compiler, R a renderer for a structured intermediate representation, G a semantic-guidance renderer, M the downstream model, and $a(x)$ a prospectively fixed inference profile. P1.1 tested replacement compilation:

$$\hat{y} = M(R(C(x)); \theta)$$

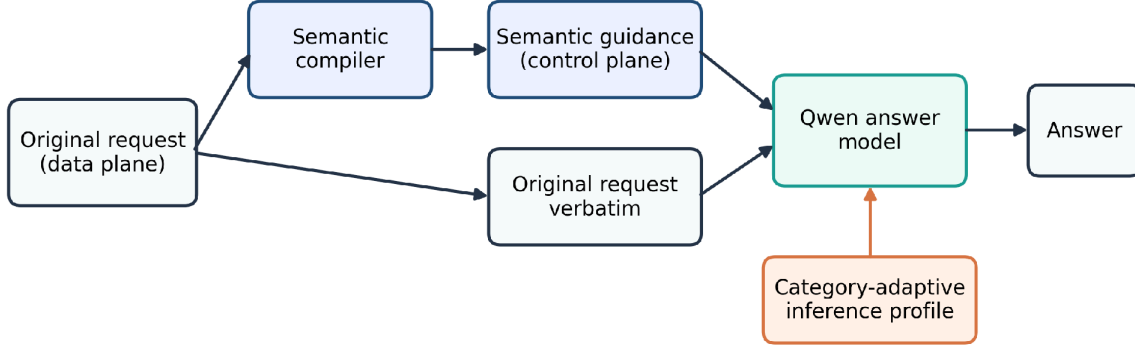
Here, compiler output replaces the original source. P1.2 and P1.3 instead tested source-preserving compound inference:

$$\hat{y} = M([G(C(x)); x], a(x); \theta)$$

The second formulation separates a control plane from a data plane. The control plane highlights the objective, active constraints, explicit precedence, ambiguity, and output requirements. The data plane preserves the complete task-bearing source: problem statements, code, formulas, tables, answer choices, and context. The category-adaptive profile selects Qwen's thinking or non-thinking configuration by benchmark category; it also changes the associated sampling parameters and maximum output length.

A Replacement compilation (P1.1)

The representation replaces the task-bearing source.

B Source-preserving compound intervention (P1.2-P1.3)

The original source remains authoritative; semantic guidance and the inference profile steer execution.

Figure 1. Evolution of the experimental architecture. P1.1 replaced the source with a structured rendering. P1.2 and P1.3 preserved the source and added semantic guidance. P1.3 prospectively confirmed the complete source-preserving augmentation plus adaptive-inference package; it did not independently establish the contribution of each component.

This distinction constrains the causal claim. P1.3 compares a multi-component package with a raw non-thinking baseline. The experiment establishes the performance of that package on the tested setting. Component contrasts help interpret the result, but they do not convert the package-level effect into proof that semantic augmentation alone was causal.

4. Experimental Program and Methods

4.1 Sequential study design

The program progressed through prospectively specified studies. Later designs were informed by earlier results, but each study fixed its benchmark, treatments, model revisions, schedules, evaluators, and analysis in version control before GPU acquisition. P1.2's full-package comparison was explicitly exploratory. P1.3 then treated the same comparison as its sole primary endpoint. Because the same tasks, stress transformations, augmentations, downstream model, and evaluators were reused, P1.3 is described as a prospective seed replication rather than an independent replication.

Study	Purpose	Downstream models	Conditions	Scheduled cases
P1 pilot	Infrastructure and semantic-compiler feasibility	Qwen3-8B	RAW, rewrite, TaskPacket replacement	192
P1.1	Standardized test of structured replacement	Qwen3-1.7B, 4B, 8B	RAW, Luna rewrite, TaskPacket replacement	1,872
P1.2	Qwen-native rewrite, source-preserving augmentation, and adaptive inference	Qwen3-1.7B, 8B	Five representation/inference conditions	2,080
P1.3	Prospective seed replication of the package-level effect	Qwen3-1.7B	RAW/AUGMENT × non-thinking/adaptive; two new seeds	768

Table 1. Experimental progression from feasibility testing to prospective seed replication.

4.2 Benchmark and semantic-stress construction

The standardized public benchmark contained 48 IFBench out-of-domain constraint tasks and 48 LiveBench tasks: eight each from coding, reasoning, mathematics, data analysis, instruction following, and language. IFBench was pinned at revision db69a6f05689830b0068b8f1529ebcfd2f3b164c and used task IDs 0-47. LiveBench was pinned to release 2024-11-25 at revision b0fb577be60ea5ba6e3e677a84af127c893af0ef; exact question IDs are listed in Appendix F. Each public task had a canonical CLEAN realization and a SEMANTIC_STRESS realization. Stress families represented scattered constraints (S1), corrections or overrides (S2), resolvable precedence conflicts (S3), reference resolution (S6), nested conditional logic (S7), and distractor-heavy multi-objective framing (S8).

P1.3 reused the exact 96 semantic-stress public tasks and frozen augmentation objects from P1.2 and excluded the clean and clarification arms. No task, transformation, or augmentation was modified after P1.2. Three tasks had unresolved augmentation artifacts; their 12 affected condition-seed rows were retained as intention-to-treat preprocessing failures.

4.3 Semantic compiler, source-preserving representations, and upstream cost

GPT-5.6 Luna served as the semantic front end in P1.1 and P1.2 [18]. The requested and returned identifier was gpt-5.6-luna; a distinct dated provider snapshot was not exposed by the model page. The frozen Responses API configuration used medium reasoning effort, a 16,384-token output ceiling, a 360-second client timeout, store=false, and truncation disabled. Across 208 source prompts, preprocessing produced 341 provider attempts, 201 accepted semantic plans, seven unresolved slots, and a measured API cost of \$0.7099. P1.1 rendered TaskPacket objects as replacement prompts. P1.2 instead generated a Qwen-oriented semantic plan, a natural-language rewrite, and deterministic source-preserving augmentation that prepended concise guidance to the complete original request. Evidence references used exact source quotations and deterministic local binding rather than model-generated character arithmetic.

P1.3 made no provider calls and reused the frozen P1.2 augmentation objects. Its seed replication therefore evaluates robustness to new downstream stochastic draws while holding compiler output fixed. The measured Luna cost is a one-time preprocessing cost for this fixed set; it is separate from the downstream GPU metrics reported below.

4.4 Models and frozen Qwen inference profiles

P1.1 used Qwen3-1.7B, 4B, and 8B. P1.2 used Qwen3-1.7B and 8B. P1.3 used only Qwen3-1.7B at revision 70d244cc86ccca08cf5af4e1e306ecf908b1ad5e. P1.3 ran on a single NVIDIA RTX 5090 with Python 3.12.3, PyTorch 2.11.0+cu128, Transformers 4.57.6, driver 580.126.09, and the official Qwen3 chat template [1,19].

The non-thinking profile used enable_thinking=false, temperature 0.7, top-p 0.8, top-k 20, and max_new_tokens 8,192. The thinking profile used enable_thinking=true, temperature 0.6, top-p 0.95, top-k 20, and max_new_tokens 16,384. Both profiles sampled rather than using greedy decoding. The adaptive policy enabled the thinking profile for LiveBench coding, reasoning, mathematics, and data-analysis tasks and used the non-thinking profile for IFBench, instruction-following, and language tasks. Thus RAW_ADAPTIVE - RAW_NT is an adaptive inference-profile contrast: it jointly changes thinking mode, decoding parameters, and output budget. P1.3 used seed roots 20260829 and 20260830; treatment identity was excluded from derived seeds so paired conditions used matched stochastic seeds.

4.5 P1.3 treatment definitions

Condition	User content	Inference profile	Role in analysis
RAW_NT	Original semantic-stress request	Non-thinking	Naive baseline
RAW_ADAPTIVE	Original semantic-stress request	Category-adaptive	Adaptive-profile effect
AUGMENT_NT	Frozen semantic guidance + original request verbatim	Non-thinking	Source-preserving augmentation effect
AUGMENT_ADAPTIVE	Identical augmented content	Category-adaptive	Primary intervention package-level

Table 2. P1.3 conditions. The primary comparison was the complete source-preserving augmentation plus adaptive-inference package versus RAW_NT.

4.6 Outcomes and statistical analysis

Public tasks were scored with pinned IFBench and LiveBench programmatic evaluators. The underlying task was the statistical unit. For each task and condition, scores were first averaged across the two new seed roots. Paired task-level effects were then estimated with 10,000 percentile-bootstrap resamples of the 96 tasks, using analysis seed 20260831 and a two-sided 95% interval. No stratification was used. Frozen preprocessing failures entered the intention-to-treat analysis as prespecified zero scores. The positive classification rule required a point estimate of at least +0.05 and an interval entirely above zero; passing that rule does not establish that the population effect itself is at least +0.05.

The primary endpoint was QWEN_AUGMENT_ADAPTIVE minus RAW_NT on the 96 semantic-stress public tasks. Secondary decomposition compared RAW_ADAPTIVE with RAW_NT, QWEN_AUGMENT_NT with RAW_NT, QWEN_AUGMENT_ADAPTIVE with QWEN_AUGMENT_NT, and QWEN_AUGMENT_ADAPTIVE with RAW_ADAPTIVE. Seed-specific estimates, complete-case sensitivity, benchmark-source subgroups, stress families, and a deterministic stress-contract audit were secondary. Secondary and subgroup intervals were not multiplicity-adjusted and should be interpreted accordingly.

4.7 Evidence custody and analysis isolation

Each GPU run recorded exact model revision, schedule hash, seed derivation, environment, raw generations, parser outputs, scorer traces, and systems telemetry. Raw acquisition was finalized and hashed before aggregate treatment analysis. Evidence was transferred to local storage and verified against remote manifests before the frozen analysis was executed. P1.3 accounted for all 768 scheduled cases: 756 real generations, 12 frozen preprocessing-failure rows, and no inference failures.

5. P1.1: Structured Replacement Was Harmful as a Universal Interface

P1.1 tested the strongest form of prompt compilation: a TaskPacket rendering entirely replaced the original request. The prospectively specified 1.7B semantic-stress contrast was -0.054 (95% CI -0.145 to 0.037; 12 wins, 64 ties, 20 losses), yielding a null or inconclusive primary classification. Effects at 4B and 8B were more negative, and replacement also harmed clean prompts across all three model sizes.

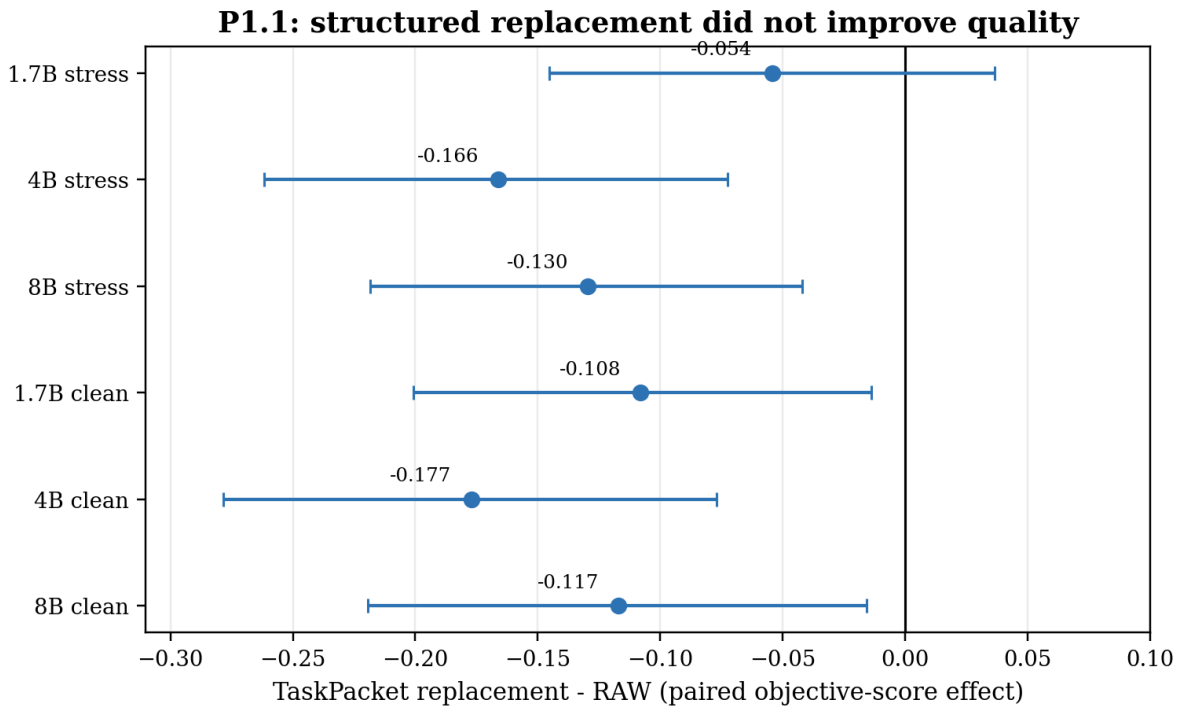


Figure 2. P1.1 paired effects of TaskPacket replacement relative to RAW. Structured replacement did not improve the primary 1.7B semantic-stress endpoint and was harmful in several secondary model and clean-prompt conditions.

The structured condition often reduced answer length and latency, but lower quality accompanied those savings. Error analysis indicated that a valid semantic schema could still omit, compress, or de-emphasize formulas, answer-choice mappings, data relationships, and other task-bearing structure. P1.1 therefore falsified the claim that semantic structure should universally replace the original prompt.

6. P1.2: Source-Preserving Augmentation and Adaptive Inference

6.1 Semantic rewriting alone

P1.2 first tested a Qwen-native natural-language rewrite. For Qwen3-1.7B on semantic-stress tasks, RAW_NT scored 0.298 and QWEN_REWRITE_NT scored 0.287, for an effect of -0.011 (95% CI -0.087 to 0.066; 13 wins, 70 ties, 13 losses). The result was null. At 8B, rewriting was significantly harmful (-0.068, 95% CI -0.138 to -0.002). Semantic rewriting therefore did not provide a reliable general-purpose improvement.

6.2 Source-preserving augmentation and category-adaptive inference

Source-preserving augmentation was more promising. At 1.7B, augmentation under the non-thinking profile improved the semantic-stress mean by +0.041 relative to RAW_NT, although the interval crossed zero. The category-adaptive inference profile improved RAW by +0.050 (95% CI 0.003 to 0.103) and improved augmented prompting by +0.071 (95% CI 0.012 to 0.135). These contrasts change the Qwen thinking switch together with its recommended sampling profile and output budget; they are not pure reasoning-only manipulations.

The full compound comparison, QWEN_AUGMENT_ADAPTIVE versus RAW_NT, was not the P1.2 primary endpoint and was treated as exploratory. It improved Qwen3-1.7B by +0.112, with an exploratory confidence interval above zero, while showing no corresponding gain at 8B. That observation motivated P1.3.

7. P1.3: Prospective Seed Replication of the Compound Intervention

7.1 Design and primary result

P1.3 prospectively specified the previously exploratory compound contrast as its sole primary endpoint. It reused the same 96 semantic-stress public tasks, the same frozen source-preserving augmentations, the

same Qwen3-1.7B revision, and the same evaluators, while introducing two new seed roots. Scores were averaged across seeds within task before statistical inference. This design tests repeated-sampling robustness rather than external generalization.

RAW_NT achieved a mean objective score of 0.272, whereas QWEN_AUGMENT_ADAPTIVE achieved 0.360. The paired effect was +0.088 (95% CI 0.018 to 0.160; 24 wins, 63 ties, 9 losses), satisfying the prespecified positive-classification rule. Relative to the RAW_NT mean, this corresponds to a 32% increase in objective score. The lower confidence bound was positive but below +0.05; accordingly, the study establishes a positive effect under its classification rule, not that the population effect necessarily exceeds the practical threshold.

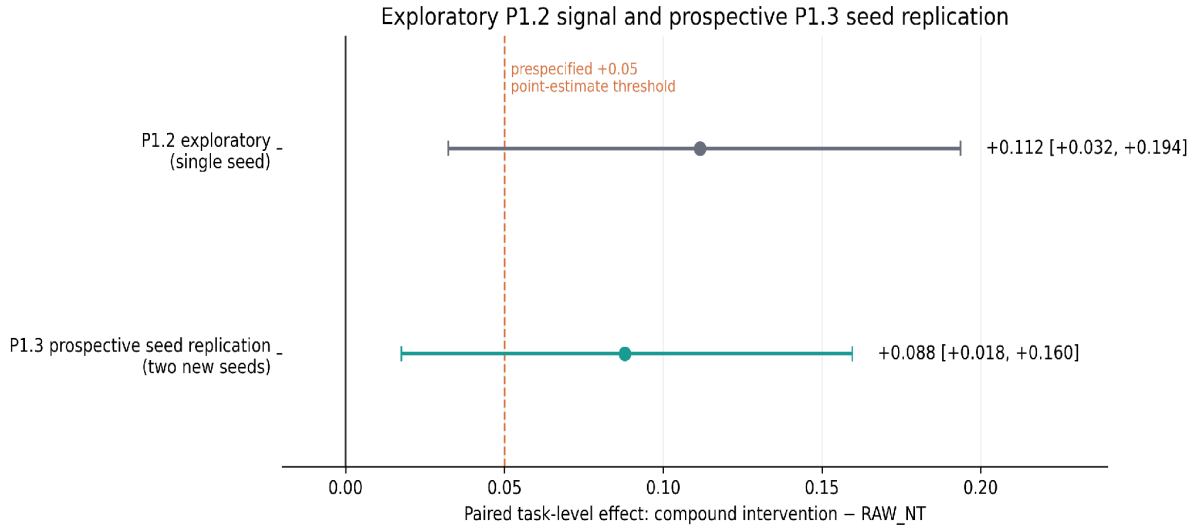


Figure 3. Exploratory P1.2 signal and prospective P1.3 seed replication. P1.3 held the tasks, augmentations, model, and evaluators fixed and introduced two new sampling seeds.

7.2 Seed robustness and sensitivity analyses

Both seed-specific point estimates were positive. Seed 20260829 produced +0.108 (95% CI 0.026 to 0.190), and seed 20260830 produced +0.068 (95% CI -0.009 to 0.147). The second seed was individually inconclusive, but the direction agreed across seeds and the task-averaged interval excluded zero.

The conclusion was robust to the pre-specified sensitivity analyses. Excluding the three tasks with unresolved augmentation artifacts increased the effect to +0.093 (95% CI 0.021 to 0.168; 93 tasks). A contract-aware secondary score, which applied machine-verifiable semantic-stress output requirements where available, yielded +0.085 (95% CI 0.016 to 0.156).

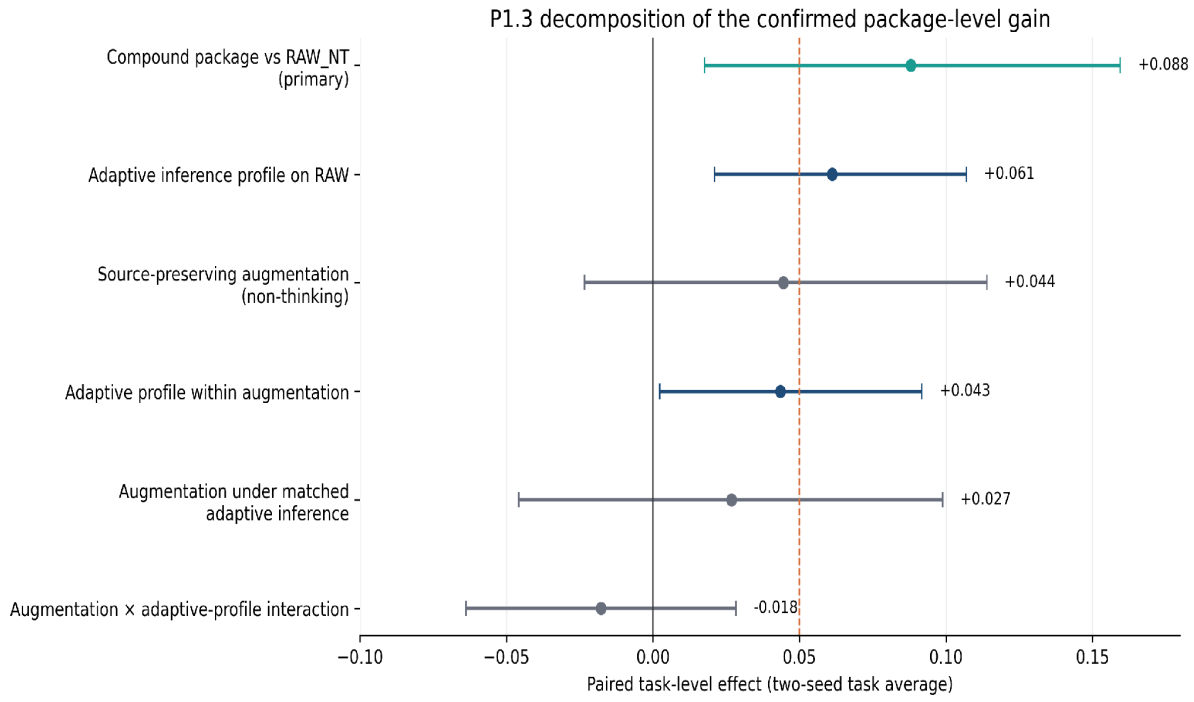


Figure 4. Decomposition of the P1.3 package effect. The category-adaptive inference profile produced a positive effect on RAW, while the incremental contribution of source-preserving augmentation under matched adaptive inference remained uncertain; the interaction estimate was not beneficial.

7.3 Component decomposition

The decomposition is central to causal interpretation. The adaptive Qwen inference profile improved RAW by +0.061 (95% CI 0.021 to 0.107). Source-preserving augmentation under the non-thinking profile improved RAW by +0.044, but the interval crossed zero. Activating the adaptive profile within augmentation added +0.043 (95% CI 0.002 to 0.092), although the point estimate remained below the frozen +0.05 practical threshold. Under matched adaptive inference, augmentation exceeded RAW by +0.027 (95% CI -0.046 to 0.099). The augmentation-by-adaptive-profile interaction was -0.018 and inconclusive.

P1.3 therefore confirms the complete package relative to RAW_NT. It does not establish semantic augmentation as the active ingredient, nor a beneficial interaction between augmentation and the adaptive profile. The matched-inference augmentation estimate is directionally positive but uncertain.

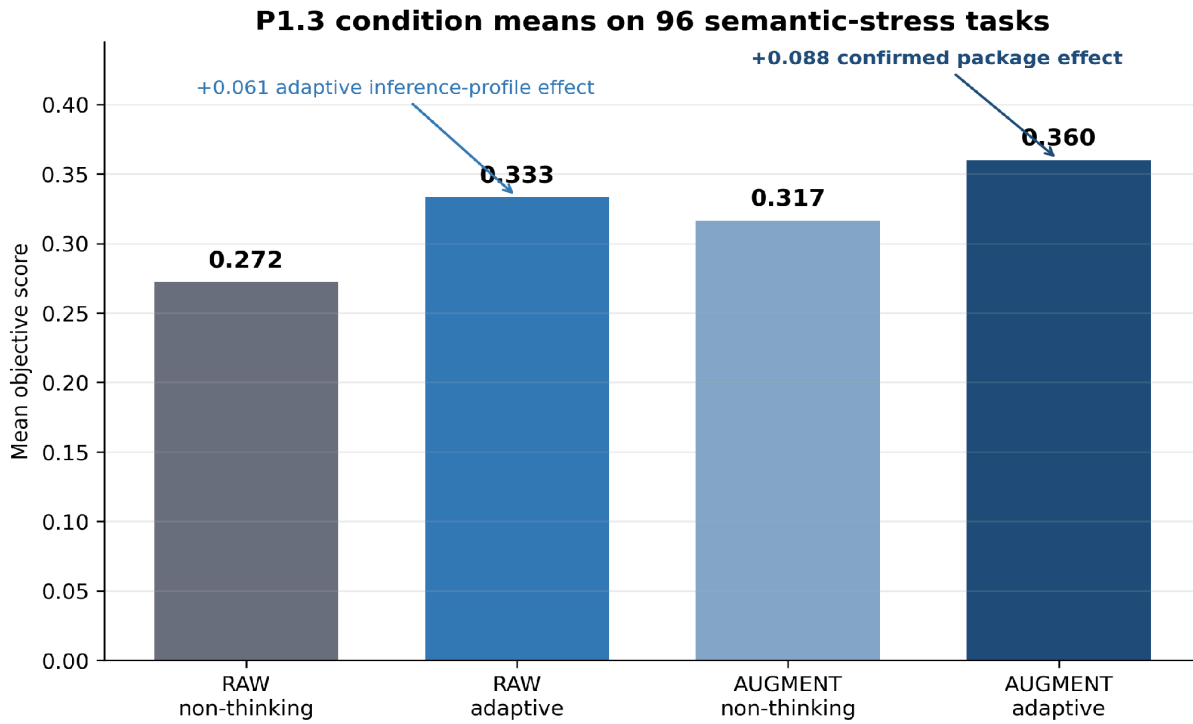


Figure 5. P1.3 condition means on the 96 semantic-stress tasks. The adaptive inference profile and source-preserving augmentation each increased the point estimate, but only the complete package versus RAW_NT was the primary confirmatory comparison.

7.4 Benchmark and stress-family effects

The package-level gain was not confined to formal instruction-following constraints. LiveBench showed a +0.103 effect (95% CI 0.019 to 0.195), whereas IFBench showed +0.073 with an interval crossing zero. Among stress families, S8 distractor-heavy multi-objective tasks produced +0.169 (95% CI 0.021 to 0.352). These were secondary analyses with small subgroups and no multiplicity adjustment.

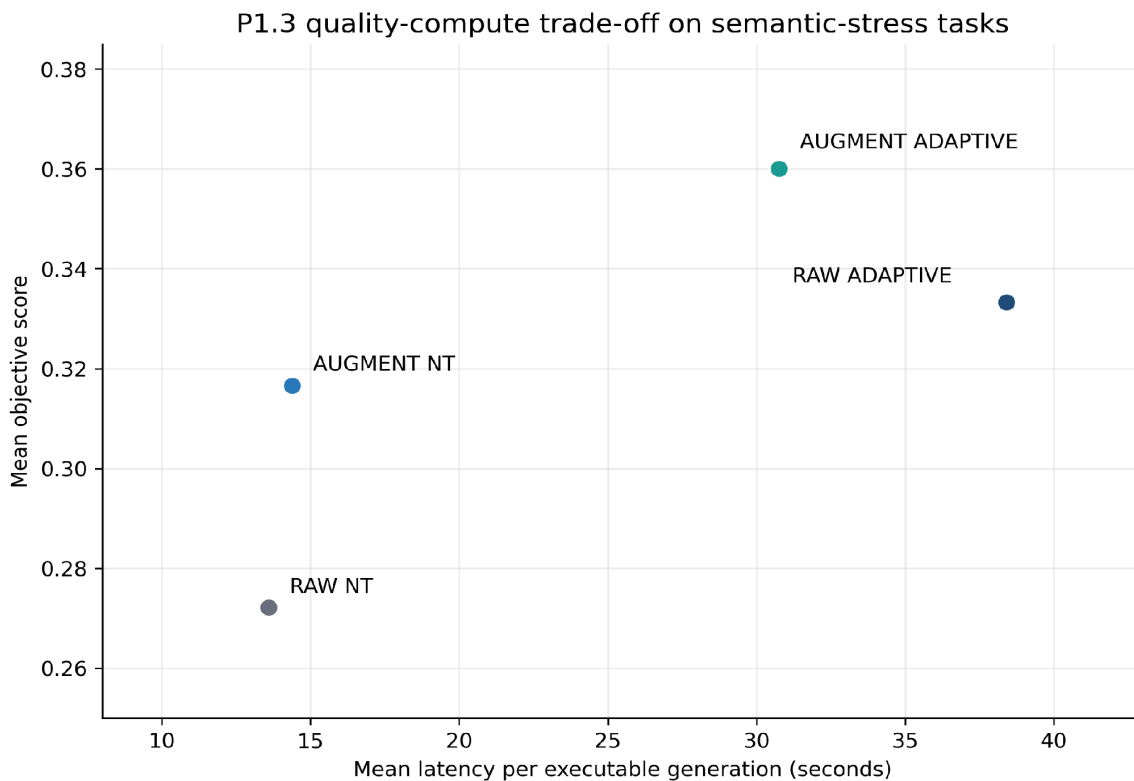


Figure 6. P1.3 quality-compute trade-off. The complete package produced the highest mean objective score but also materially higher latency than the raw non-thinking baseline.

7.5 End-to-end cost and systems behavior

The confirmed quality gain required substantially more downstream computation. Among executable generations, QWEN_AUGMENT_ADAPTIVE used a mean of 746 input tokens versus 465 for RAW_NT and produced an average of 1,905 total output tokens, including 1,184 thinking tokens. Mean latency increased from 13.6 s to 30.8 s, and total GPU time increased from 2,609 to 5,721 s. Peak VRAM remained essentially unchanged because the downstream model was the same.

Upstream semantic compilation added a separate one-time cost. P1.2 processed 208 prompts through GPT-5.6 Luna using 341 provider attempts, accepted 201 artifacts, left seven unresolved, and incurred \$0.7099 in measured API cost. P1.3 reused those frozen artifacts and made no provider calls. A production system would need to account for compiler latency, cost, and failure rate in addition to downstream inference.

Condition	Input tokens mean	Thinking tokens mean	Final / total output mean	Latency mean (s)	GPU seconds total
RAW — non-thinking	465.17	0	842.31 / 843.14	13.59	2,608.80
RAW — adaptive	463.83	1,809.20	565.22 / 2,376.32	38.40	7,371.94
AUGMENT — non-thinking	747.16	0	889.11 / 889.72	14.39	2,676.72
AUGMENT — adaptive	745.91	1,184.41	718.63 / 1,904.80	30.76	5,720.50

Table 3. P1.3 systems metrics for Qwen3-1.7B. Means use executable generations: 192 cases for each raw condition and 186 for each augmented condition. Six frozen augmentation-preprocessing failures per augmented condition generated no GPU work. GPU-seconds are observed execution totals.

8. Interpretation and Causal Scope

8.1 What the evidence establishes

The sequential studies support a narrow hierarchy of claims. Structured replacement is harmful as a universal answer-model interface. Qwen-oriented rewriting alone is not a reliable intervention. The complete source-preserving augmentation plus category-adaptive inference package improves Qwen3-1.7B on the fixed 96-task semantic-stress population under two new sampling seeds. The independent semantic-augmentation contribution under matched inference remains unconfirmed.

The P1.3 result is unlikely to be a single-seed accident: both seed estimates were positive, complete-case analysis was slightly stronger than intention-to-treat, and contract-aware scoring preserved the conclusion. However, because P1.3 reused the same tasks, stress transformations, augmentations, model, and evaluators, it is a prospective repeated-sampling confirmation rather than an independent replication or generalization test.

8.2 Control plane, data plane, and inference profile

The results suggest that semantic analysis is better treated as a control plane than as a replacement for the task. The compiler can highlight active constraints, precedence, and output requirements, while the original source remains authoritative. This distinction is especially important for mathematics, code, and data-analysis tasks in which formulas, tables, or answer mappings are difficult to summarize safely.

The adaptive inference profile is a second control lever. Its positive RAW_ADAPTIVE contrast shows that category-based changes to thinking mode, sampling, and output budget improve performance without semantic augmentation. The complete package combines both mechanisms productively on average, but the interaction estimate was not positive and the augmentation increment under matched adaptive inference remained uncertain.

8.3 Open causal questions

The current experiment does not identify the cheapest or minimal cause of the package-level effect. Missing controls include direct Luna answers, deterministic or shuffled guidance, Qwen self-planning, always-thinking Qwen, and equal-latency or equal-cost larger-model baselines. P1.2 suggested partial

recovery of the raw 1.7B-to-8B score gap, but P1.3 did not repeat the 8B arm; model-scale substitution therefore remains exploratory.

8.4 Toward selective compound systems

A practical architecture should not apply the same intervention to every request. Earlier clean-arm results show that unnecessary compilation can be harmful. A lightweight controller should estimate semantic difficulty and expected value of added compute, bypassing simple requests, using source-preserving guidance for constraint-heavy inputs, and enabling the expensive profile only when its expected benefit justifies the cost.

9. Limitations

The confirmatory result is limited to Qwen3-1.7B. It does not establish generalization to Llama, Gemma, Mistral, proprietary models, or models trained specifically to consume control structures. It also reuses one fixed 96-task population, the same researcher-authored stress transformations, and the same frozen augmentations. New stochastic draws strengthen internal robustness but do not test new tasks, new compiler outputs, independent researchers, or another model family.

The semantic compiler is a proprietary API model. The requested and returned identifier was gpt-5.6-luna, but no distinct dated provider snapshot was exposed. The study reports the exact API configuration and measured cost, yet future provider drift cannot be eliminated. P1.3 itself made no provider calls because it reused frozen outputs. Direct Luna answering and cheaper non-model controls were not included.

The primary package changes more than one component relative to RAW_NT: it adds source-supported guidance and activates a category-adaptive inference profile that changes thinking mode, sampling parameters, and output allowance. P1.3 confirms this complete intervention. It does not independently confirm semantic augmentation, a beneficial interaction, or a reasoning-only causal mechanism.

The result met the prespecified positive-classification rule because the point estimate exceeded +0.05 and the percentile-bootstrap interval excluded zero; the lower bound of +0.018 does not establish that the population effect exceeds +0.05. Secondary and subgroup intervals were not adjusted for multiplicity. Some stress-added output requirements are only partially represented by the official scorer, although the prospectively frozen contract-aware sensitivity remained positive. The quality gain also required substantially greater latency and GPU time, so no quality-cost Pareto improvement is claimed.

10. Conclusion

Semantic preprocessing is not uniformly beneficial. Replacing a natural-language request with the current structured intermediate representation degrades Qwen3, and natural-language rewriting alone does not reliably improve quality. The most promising design preserves the source and treats compiler output as additional control information rather than a substitute for task data.

A prospective seed replication confirms that, on the tested fixed benchmark and Qwen3-1.7B revision, the compound package of source-preserving semantic augmentation and a category-adaptive inference profile improves objective performance by 0.088 points (95% CI 0.018 to 0.160). The result is robust to two new seed roots and prespecified sensitivity analyses, but the independent semantic-augmentation contribution remains uncertain and the gain requires additional test-time computation. The next scientific step is a fresh-task and cross-model replication with controls that isolate augmentation from decoding profile and inference-time compute.

Data and Materials Availability

Frozen prompts, source-preserving augmentation objects, raw generations, evaluator traces, and analysis scripts are retained in versioned research archives and are available from the author to qualified researchers upon reasonable request, subject to benchmark licenses and provider terms. The manuscript does not claim public preregistration or independent timestamping; experimental conditions were prospectively specified and committed in the project repository before acquisition.

Author Contributions

Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, visualization, project administration, writing - original draft, and writing - review and editing: Thibaud Peverelli.

Competing Interests

The author declares no competing interests.

References

- [1] Yang, A., et al. (2025). Qwen3 Technical Report. arXiv:2505.09388.
- [2] Kandogan, E., et al. (2024). A Blueprint Architecture of Compound AI Systems for Enterprise. arXiv:2406.00584.
- [3] Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176.
- [4] Ong, I., et al. (2024). RouteLLM: Learning to Route LLMs with Preference Data. arXiv:2406.18665.
- [5] Khattab, O., et al. (2023). DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. arXiv:2310.03714.
- [6] Yang, C., et al. (2023). Large Language Models as Optimizers. arXiv:2309.03409.
- [7] Zhang, T., Yuan, J., & Avestimehr, S. (2024). Revisiting OPRO: The Limitations of Small-Scale LLMs as Optimizers. arXiv:2405.10276.
- [8] Zhu, K., et al. (2023). PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. arXiv:2306.04528.
- [9] Zhou, J., et al. (2023). Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911.
- [10] Pyatkin, V., et al. (2025). Generalizing Verifiable Instruction Following. arXiv:2507.02833.
- [11] White, C., et al. (2025). LiveBench: A Challenging, Contamination-Limited LLM Benchmark. arXiv:2406.19314v2. <https://arxiv.org/abs/2406.19314>
- [12] Wei, J., et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.
- [13] Snell, C., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally Can Be More Effective Than Scaling Model Parameters. arXiv:2408.03314.
- [14] Madaan, A., et al. (2023). Self-Refine: Iterative Refinement with Self-Feedback. Advances in Neural Information Processing Systems, 36. arXiv:2303.17651.
- [15] Ho, N., Schmid, L., & Yun, S.-Y. (2022). Large Language Models Are Reasoning Teachers. arXiv:2212.10071.
- [16] Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., & Zou, J. (2024). Mixture-of-Agents Enhances Large Language Model Capabilities. arXiv:2406.04692.
- [17] Chen, L., et al. (2025). Optimizing Model Selection for Compound AI Systems. arXiv:2502.14815.
- [18] OpenAI. (2026). GPT-5.6 Luna model documentation. <https://developers.openai.com/api/docs/models/gpt-5.6-luna> (accessed 29 August 2026).
- [19] Qwen Team. (2025). Qwen3-1.7B model card. <https://huggingface.co/Qwen/Qwen3-1.7B> (accessed 29 August 2026).

Appendix A. Claim Status and Evidentiary Boundary

Claim	Study / analysis	Estimate	Status
TaskPacket replacement improves 1.7B stress quality	P1.1 primary	-0.054 [-0.145, 0.037]	Not supported
Qwen-native rewrite improves 1.7B stress quality	P1.2 primary	-0.011 [-0.087, 0.066]	Null / inconclusive
Adaptive Qwen inference profile improves RAW 1.7B	P1.3 secondary	+0.061 [0.021, 0.107]	Supported
Complete package improves RAW_NT on the fixed 96-task set	P1.3 primary	+0.088 [0.018, 0.160]	Confirmed package-level effect
Semantic augmentation adds value under matched adaptive inference	P1.3 secondary	+0.027 [-0.046, 0.099]	Not established
Augmentation has a beneficial interaction with the adaptive profile	P1.3 secondary	-0.018 [-0.064, 0.028]	Not supported
Effect generalizes across small-model families	External validity	Not tested	Not established
Compound intervention substitutes for model scale or improves Pareto frontier	Cross-scale / systems claim	Not tested confirmatorily	Not established

Table A1. Principal claims and evidentiary status. The confirmed package-level effect is separated from component and generalization claims.

Appendix B. Exact Model, Runtime, and Case Accounting

Item	P1.1	P1.2	P1.3
GPU	RTX 5090	RTX 5090	RTX 5090
Driver	580.126.09	580.126.09	580.126.09
PyTorch / CUDA runtime	2.11.0+cu128 / 12.8	2.11.0+cu128 / 12.8	2.11.0+cu128 / 12.8
Transformers	4.57.6	4.57.6	4.57.6
Qwen3 revisions	1.7B / 4B / 8B (pinned)	1.7B / 8B (pinned)	1.7B: 70d244cc86cc...
Scheduled cases	1,872	2,080	768
Real generations	1,776	2,038	756
Preprocessing-failure rows	96	42	12
Inference failures	0	0	0
Acquisition duration	2 h 49 min	12 h 55 min	5 h 10 min

Table B1. Frozen model, runtime, and case-accounting details.

Appendix C. Reproducibility and Evidence Custody

P1.3 was prospectively frozen in version control at commit 045c44891cc5d52434981b7ce822ff7cbcb1647 before GPU acquisition. The confirmatory run was EXP-P01-P13CONF-001 / P13-RUN-045c44891cc5-7395027dbd8e-S20260829-20260830. All 768 scheduled rows were accounted before analysis. The frozen hashes identify exact artifacts but do not, by themselves, constitute independent public timestamping.

Pre-analysis journal SHA-256: 26b046d21c1ccd94ec5073e5397859a541e16566bbb54b48e3deb414101b17be

Freeze-record SHA-256: 3afc338887acd07c9519b5c28c1052360c495af6a8700c83be20cf5c7e3d0f40

Evidence-manifest SHA-256: 35bc4353f9fa310a9fc453de77f105e7f0f3eb1d829618baf37cf6f856134557

Acquisition-archive SHA-256: 28baf35ddb1bf0dea905fb5618268a6950611493e7fd51dad20744b4ad30241e

Official-analysis SHA-256: ea6027097c7a736825827e32f4821428ba041d819e9e019dac0d8c93cb495c04

Result-manifest SHA-256: 50ca85737d9c07e1e167eb05732ab249d2e39b71f25bd2c23783009c439a2249

Raw evidence was frozen before treatment analysis, transferred from the GPU environment, and verified against remote manifests. The statistical analysis was reproduced from the frozen case rows. No provider

call occurred during P1.3 acquisition, and no scientific case was rerun after outputs were observed. Materials are available from the author upon reasonable request, subject to applicable licenses and terms.

Appendix D. Benchmark and Evaluator Caveats

The experimental program revised evaluator defects only prospectively. The initial pilot's numeric scorer sometimes extracted an intermediate number rather than the final answer, and its code-format checks were overly literal. Later studies used corrected deterministic evaluation and pinned official public scorers.

Some semantic-stress transformations add output requirements that are not fully represented in the corresponding official benchmark scorer. P1.3 therefore retained official scoring as the primary endpoint and added a prospectively frozen StressContractAudit for 33 machine-verifiable tasks. The contract-aware package effect remained positive, but the audit does not cover every transformation.

Formal IFBench scores measure verifiable instruction compliance and should not be interpreted as a complete measure of holistic answer quality. LiveBench category and stress-family subgroups contain small samples and are secondary. A further clarification-evaluator edge case distinguishes explicit requests for missing data from safe abstention without a request; clarification results are not part of the 96-task public primary endpoint.

Appendix E. Representative Source-Preserving Interventions

The examples below are descriptive and were not used to define the statistical analysis or exclude cases. Excerpts are shortened to show the intervention structure without reproducing complete benchmark prompts.

E.1 Coreference and positional constraint - successful intervention

Source excerpt: *“The second word ... and the second-to-last word ... should be the word whisper.”*

Guidance excerpt: *Treat “it” as the complete reference-block request; enforce both positional-word constraints; preserve the supplied reference text; do not infer an unsupported transformation.*

Observed pattern: the augmented conditions satisfied the formal constraint under both seeds, while both raw conditions failed. This case illustrates how source-supported control information can clarify reference resolution and exact output requirements.

E.2 Creative pronoun-count constraint - unsuccessful intervention

Source excerpt: *“Write a ... screenplay of a medieval king ... [and] include at least 39 pronouns.”*

Guidance excerpt: *Produce the requested screenplay; retain the medieval/proto-vaporwave premise; include at least 39 pronouns; output only the screenplay.*

Observed pattern: RAW passed under both seeds, whereas the augmented conditions failed the verifier. This counterexample shows that source preservation does not guarantee neutral guidance or successful compliance; the intervention can still alter salience and generation behavior.

E.3 Intervention schema excerpt

The semantic front end produced a closed QwenPromptPlan containing the task type, objective, critical constraints, precedence rules, conditional rules, output contract, ambiguities, blocking and non-blocking unknowns, reasoning recommendation, and source-bound evidence quotations. The downstream augmented prompt contained a concise rendered guidance block followed by the immutable original request exactly once. TaskTruth and official answers were not included in provider inputs or downstream prompts.

Appendix F. Benchmark Selection and Frozen Identifiers

IFBench was pinned at revision db69a6f05689830b0068b8f1529ebcfd2f3b164c. The selected OOD-constraint task identifiers were 0 through 47 inclusive, selected deterministically before model acquisition.

LiveBench was pinned to release 2024-11-25 at revision b0fb577be60ea5ba6e3e677a84af127c893af0ef. Eight objectively scoreable tasks were selected from each category using canonical question-ID order. Exact identifiers follow.

Category	Frozen LiveBench question identifiers
Coding	009f47e8275c73de2eae601d76ff7163f613880788d870c1484b1c2a1383fbd 0119bdfb9e79f05e988b6d3408e2fdd0b7e4bc11e81ff9f24bed96268c503801 01c73e7f5bd7b486915f97829994aca9b95d6116c510d9f3781b5399dea066bd 030471ea2fc50a29f82560555033ceaec6b127c9ad0f68d34565ba221279d54b 03741bcada4e970eedeb7f28fd45c496779b98f3df8e513f992a51cf072f67c 042b958543e9a61fd083dfd6067a91845b8e272295a282512ec9b527cf2e113 049ed4055f772aa9bccfda47ae425ee8a3c16d2c74330818d0397890795ea7fb 0664f46bd7bf6217d229be6762834af2aa642e7b97b14469f478c93a382ee520
Reasoning	0405cc3d80d24b05c54ae0284daac12f64b8ec440dfc3ab8e35b0970f414e91e 04207d46453963ade715fc168f3ef781edc3f5355fc03090911cb355fb73867c 09719ca5e81c332df5d59ed37737c7351b5d68568933f4e7bbe08f0b22f2d030 0b02d4ab897f5f7eae1f5715f159ba9309ae8e91ce0403d8df960db77253e36c 0daa7ca38beec4441b9d5c04d0b98912322926f0a3ac28a5097889d4ed83506f 0f6ad553bbe8ca60dfd629dccb15f12a4891830b6e5eec2cdf64ee2eccebc38 0f798486e95949a2e4b03b330f24fa5a77c232aa3d9549bb4233aed5ceccca89 115ee3f4cecb92009f145dd6ec969d05ce0b255f8a8a4d8bb686253be96aafc
Math	01540b9b0b068942c8adeec17f825f0a9a3b9f52a1f61704bfebbb4c4060647 017a695f2234c3e36c6fc73d6764b0f713ec9bdebd07f021d63f59a5d2c07181 01fa9af2dad55a3be29d1fe93f385ad2acade52ed71c6af3dd6f65c9da7ba142 0499deda2f068008d488551abf96b4b758c6ed6b79cd2ec6a204d1250b140421 071c68cbb1afe46565f2dcfad6df4a077cb7c54fb2232c8ff65057657d2017 07a0d7af7b149f35d0441e3b732fc4706ac286fca96748d7c3e4ceb95af46558 0ae7ae87b394fd2199a6d3979637650610c73f74767bd64fc58acaf711efe061 0d3a48049dc5da31531654f59e7866832579f812b892610bd1b178fcd4010d0c
Data Analysis	00d7be878c842d12814cb113caf8503525f8cf845b7d3ca4b8387c843f06bec9 010107031fd6e54fcac08ac186ff4f0a9018d887bf5920f921f14e80bd82633 01fc14e123214c67cbf235824d1ec952a825d57f8464ecc18fb9609c2781f50c 04ba0a2b8fe86cdd255723961356723f6de221cbe6bbc7af4b9ac93d45cd40ec 051ed5edf44bb798385076a1260de95b272dd2e0f5167dc78e514ce434af3ef6 0764131eaf30bb8af36ad749f144da01c0113b1cee00092dde2919287df2ba78 07844eb9fb31c8e9cac12e29662d01c221f762a67418b020ff4eae637065539a 083282355242eb434e4c4559eabea700f94928fd2e1d0d4df6a59ee143866e60
Instruction Following	006d27d0a338a42a35e7d50e01bfd5b8e502897097c15ab7a4b01a6e38c6bf4b 02b82146fcd90e18dad8f8ef85bd5566a2c7540429f8e03c73b98dac3f58cf4 0503923535fbd8badfb2ccf38989b36a8ecaa2eda332a4b515bb13932421fecf 05dd93ecef22fdd5d3a5d396616c472436ffcebf9a9c0d95fedda3ff661a11fa 0996be1780ea1ac0e68d98e0bed513b23ec11b212c3461884e28c3a752a00efc 0a0385df33f3d95602ab7f1a78de0bfc276cd8067d27336188e84341cdd02032 0b041d87e0bba1a5db140a69ef8fd4e844f1f0f0b86be7a6ceba5f9edf36d584 0cf5814b25d7aa0420a7d6d8f03e0ccd977c61fb996be04f8c3221153ba1f005
Language	032e5f4f783412d63296cc024f9b72799491127b20cbccafe1f8430091cd16ea 03ce31e3890bffe51d1fea02c6266b9094b164f2832c5afe5f2717b5834b01c1 0becbf34d783f6150de35cbf89625935cd947959ad3fc122e0da5e22e5514187 0d62cddf1b9b189b3806f8c8592cc68955cc4253d18d55e537a2cf5b14aa4830 0eb40998f098687b56ca73995efe43925a8e557b7fc49deddc648a8a2141d3ef 101e363752a4f40a813226b9d55eaeaba40a95d984f9158a12fa144bc2696733 11aaf19be1a0512eb0a266d0bcf1d639d05cf0ca75e07dbb8dc03f71f4cfb8 11fe6e706f1d052a38c42d39c3d2ef5f94a6bbbee363a0b297ffda502b2e5c6bf

Selection and scorer wrappers were frozen before downstream model acquisition. Exact prompts, transformations, and scorer traces are available from the author upon reasonable request, subject to benchmark licenses.