Cognitive Orientation Prompting Compresses LLM Output Tokens by 75% on GPQA Diamond

A Preprint

Matthew M. Gaba
Unnma LLC
Miami, FL
teo@unnma.ai

May 2026

Abstract

We present Cognitive Orientation Prompting (COP), a structured prompting framework that reduces large language model completion tokens by reorienting the model's cognitive stance rather than instructing behavioral changes. On GPQA Diamond (198 graduate-level science problems), COP compresses mean completion tokens 75% (10,366 to 2,590) with an effect size of $d = 2.3$, while reducing wall-clock latency from 201s to 55s and eliminating timeout failures (16.8% null timeout rate to 0%). The compression is content-dependent: a word-salad control of comparable length compresses only 23%, ruling out prompt-length as the explanatory mechanism. Four structurally distinct articulations of the cognitive orientation converge to within 1.6 percentage points of compression and accuracy. The effect is complexity-dependent on Qwen 3.5 27B: COP compresses across all four benchmarks tested, with magnitude scaling monotonically with baseline verbosity — from −9.8% on HumanEval (1,038-token baseline) through −23.5% on MT-Bench, −63.6% on COP-Hard, to −75.0% on GPQA Diamond (10,366-token baseline) — a log-linear fit of $R^2 = 0.97$. A 50-token brevity-gate appendix (DH+B) extends compression further across every benchmark tested. Accuracy on GPQA Diamond drops 11 percentage points (88.8% to 77.9%); we report a structural failure-mode analysis showing the cost concentrates on problems whose correct answer requires the model to override a strong default interpretation by holding alternatives open. The effect generalizes across five models spanning four families, with compression magnitude scaling with baseline verbosity. A production-realistic trial on claude-opus-4 ($N = 96$ paired samples across coding and customer-service workloads, judge-scored on six axes) replicates the compression direction, cuts mean per-prompt latency by 63.5% (31.6s → 11.6s, paired $d = 0.95$), and additionally shows quality *improvements* on all six judge axes relative to a baseline matched on prompt structure and model configuration. The standard prompt-engineering control compresses only 16% on GPQA Diamond and shows quality degradation on the production benchmark, indicating COP is doing something distinct from a brevity instruction.

Keywords system prompt design · inference optimization · token efficiency · large language models · reasoning compression · inference-time intervention

1 Introduction

Large language models have demonstrated capabilities across reasoning, coding, and analytical tasks. A persistent challenge in deployment is inference cost, which scales linearly with completion tokens, and wall-clock latency, which scales similarly. At industrial scale, even modest reductions in completion length translate to substantial cost savings, throughput improvements, and reduced latency at the inference frontier.

The prompt-engineering literature has produced techniques that improve output quality at the cost of increased token generation: chain-of-thought prompting (Wei et al., 2022), tree-of-thought reasoning (Yao et al., 2023), self-consistency decoding (Wang et al., 2023), step-back prompting (Zheng et al., 2024), and self-reflection methods (Madaan et al., 2023; Shinn et al., 2023). These techniques share a property: they improve quality by *increasing* the number of tokens generated. The dominant paradigm assumes a direct tradeoff between quality and computational cost.

We present Cognitive Orientation Prompting (COP), a structured prompting framework that intervenes at the system-prompt boundary by reorienting the model's cognitive stance rather than instructing behavioral changes. COP targets token *reduction* on hard reasoning tasks while preserving or improving output quality on production-realistic workloads. The user query is submitted unchanged; there is no prompt transformation, no input compression, and no external reranking. The intervention surface is the system prompt.

The specific construction of the COP prompt is the subject of a forthcoming patent application and is treated as proprietary; this paper reports its measured behavior, not its content. What we can disclose is the category: COP is a structured framework drawn from contemplative-philosophical traditions. The framework is domain-general (it does not specialize per task), does not name a target output length or response format, and is not a behavioral instruction. Its mechanism operates at the level of the model's *stance* toward its own generation rather than at the level of explicit rules. The specific axes along which the orientation acts, and the specific text that articulates them, are described in the pending patent application.

Six findings motivate the full presentation:

1. **COP compresses 75% on GPQA Diamond, and the compression is content-dependent.** On 198 graduate-level science problems (Qwen 3.5 27B), COP reduces mean completion tokens from 10,366 to 2,590 — a 75% compression with $d = 2.3$. A word-salad control of comparable token length compresses only 23%, establishing that the effect is not an artifact of prompt length. The remaining ∼52 percentage points of compression require explanation beyond system-prompt verbosity.
2. **COP cuts wall-clock latency 73% on GPQA Diamond and 63% on a production-realistic claude-opus-4 trial.** On GPQA Diamond (Qwen 3.5 27B), mean wall-clock latency falls from 201s under null to 55s under COP ($d = 2.51$), and timeout rates fall from 16.8% to 0% — the model under COP never generates enough tokens to exhaust the wall-clock or token budget. On the production-realistic trial described in §5 (claude-opus-4, 48 prompts × 2 reps × 2 conditions across coding and customer-service workloads), mean per-prompt latency falls from 31.6s under baseline to 11.6s under COP — a 63.5% reduction (paired $d = 0.95$), with COP faster on all 48 prompts and per-prompt latency savings ranging from 32.9% to 86.1%. The two latency findings together — academic-benchmark plus production-realistic — establish wall-clock latency reduction as a robust deployment outcome rather than a benchmark artifact.
3. **The effect is complexity-dependent on the primary model.** On Qwen 3.5 27B, COP compresses across all four benchmarks tested, with compression magnitude scaling monotonically with baseline verbosity: −9.8% on HumanEval (1,038-token baseline), −23.5% on MT-Bench (2,559-token baseline), −63.6% on COP-Hard (7,361-token baseline), and −75.0% on GPQA Diamond (10,366-token baseline). The cross-benchmark log-linear fit is $R^2 = 0.97$ ($p = 0.014$). A 50-token brevity-gate appendix (DH+B) extends compression further: −39.5% on HumanEval, −43.6% on MT-Bench, −76.0% on COP-Hard.
4. **A production-realistic trial replicates compression and shows quality improvement.** On claude-opus-4 across coding and customer-service workloads ($N = 96$ paired samples, Gemini-judged on six quality axes), COP compresses 26% (median $K = 1.35$) while improving quality on all six axes — correctness, clarity, reasoning quality, structural organization, focus, and length appropriateness. A "be-concise" control achieves greater compression (49–53%) but at the cost of correctness (−0.23 on the correctness axis). The mechanism produces a different distribution of what gets cut than a terseness instruction does.
5. **The accuracy cost on GPQA Diamond is structurally concentrated, not uniform.** COP loses 11 percentage points of accuracy on GPQA Diamond (88.8% → 77.9%). A per-problem read of the failure set identifies a single shared structural feature: the failures concentrate on problems whose correct answer requires the model to override a strong default interpretation, by holding alternatives open and verifying which is correct. On problems without this structure, COP preserves accuracy. The cost is therefore a property of a specific cognitive operation COP short-circuits, not a property of problem difficulty in general.
6. **Four structurally distinct articulations converge.** Four variants of the COP framework — differing in section structure, prose register, and combinatorial composition — produce identical compression

(75.0%–76.6%) and identical accuracy (77.9%–79.5%) on GPQA Diamond. This convergence, combined with the word-salad result, constrains the active mechanism to the *content* of the cognitive orientation rather than to specific wording.

Contributions. (1) We demonstrate large-effect-size token compression on a standard graduate-level reasoning benchmark. (2) We report wall-clock latency reductions on both the academic benchmark (73% on GPQA Diamond) and a production-realistic workload (63.5% on claude-opus-4), plus elimination of model timeouts, as first-class deployment outcomes. (3) We establish content-dependence via a word-salad control, ruling out the parsimonious null hypothesis of "any long system prompt compresses output." (4) We characterize the complexity-efficacy gradient empirically across four benchmarks on a single primary model ($R^2 = 0.97$). (5) We report a structural failure-mode analysis identifying where the accuracy cost concentrates. (6) We validate the effect across five models spanning four families. (7) We report a production-realistic trial in which COP improves judge-scored quality on all six axes while still compressing. (8) We demonstrate cross-articulation convergence, constraining the mechanism to prompt content rather than specific wording.

2 Related Work

2.1 Prompting Techniques That Increase Token Cost

Chain-of-thought prompting (Wei et al., 2022) improves reasoning by eliciting intermediate steps at the cost of increased generation length. Tree-of-thought (Yao et al., 2023) explores multiple reasoning paths with backtracking. Self-consistency (Wang et al., 2023) samples multiple chains and takes majority vote. Step-back prompting (Zheng et al., 2024) adds an abstraction pass. Self-reflection methods (Madaan et al., 2023; Shinn et al., 2023) introduce iterative refinement cycles. All of these techniques increase tokens in service of improved quality. None targets token reduction as a primary objective.

2.2 Prompt and Input Compression

A separate line of work compresses the *input*. LLMLingua (Jiang et al., 2023) removes low-contribution prompt tokens. AutoCompressor (Chevalier et al., 2023) compresses long contexts into summary vectors. Gisting (Mu et al., 2023) learns virtual tokens that compress instructions. These approaches reduce input cost. COP reduces *output* cost. The two are complementary.

2.3 Test-Time Compute Scaling and Overthinking

Recent work on "overthinking" in reasoning models is directly relevant. Chen et al. (2024) find that longer reasoning traces do not always improve accuracy in o1-class models. Muennighoff et al. (2025) introduce budget forcing — truncating thinking on easy problems without sacrificing accuracy. Snell et al. (2024) show that optimal test-time compute allocation is problem-dependent. COP can be understood as a soft form of budget control: not truncating mechanically, but orienting cognition so that shorter reasoning chains emerge naturally on problems where the model has sufficient knowledge for direct resolution.

2.4 System Prompt and Role Engineering

Salewski et al. (2023) show that expert role prompts improve reasoning. Reynolds and McDonell (2021) explore prompt programming as a discipline. Existing work on system prompts focuses on output *quality* or behavioral conformance. We are not aware of published work that systematically studies system prompt content as a mechanism for reducing output *token count* while characterizing the accuracy boundary.

2.5 Standard Benchmarks

GPQA Diamond (Rein et al., 2023) is a 198-question graduate-level science benchmark designed to be Google-proof; even domain experts with internet access achieve only ∼65% accuracy. MT-Bench (Zheng et al., 2023) provides 80 conversational problems across 8 categories. HumanEval (Chen et al., 2021) provides 164 code-generation problems.

3 Method

3.1 The COP Framework

COP is implemented at the system-prompt boundary. A fixed text block — referred to as DH in the experimental conditions below — is concatenated with any task-specific base instruction and submitted as the system prompt; the user query is then submitted unchanged. No prompt rewriting, no input compression, and no external reranking is performed.

The DH text is approximately 2,800 tokens. Its category is a contemplative-philosophical orientation framework: it reframes the model's relationship to its own token generation as a domain-general cognitive posture rather than as a per-task instruction. It contains no rules about brevity, output length, response format, or response structure. The specific construction of the framework — its sectional structure, the conceptual axes it engages, and the prose articulating them — is treated as proprietary and is not disclosed in this paper. The paper reports the framework's measured behavior across multiple structurally distinct articulations (§4.3) and characterizes its empirical envelope rather than its content. Section 4 establishes through controls that the active mechanism is the *content* of the orientation rather than the *length* of the system prompt or any specific surface wording.

The conditions tested in this paper are:

Table 1: Experimental conditions.

Code	Condition	Description
Null	No system prompt	Only the base task instruction ("Solve the following problem.").
PE	Standard prompt engineering	A short (∼150-token) behavioral instruction in the spirit of standard prompt-engineering practice: decompose, look for patterns, prefer simpler solutions, check reasoning, present clearly.
COP	Cognitive Orientation Prompting	The DH framework (∼2,800 tokens) prepended to the base instruction. On some conditions, a 50-token "brevity gate" (B) is appended; this is noted explicitly.
WS	Word-salad control	Approximately 2,800 tokens of semantically incoherent text matched to DH in length but containing no coherent orientation or instructions.

3.2 Variant Robustness

Four structurally distinct articulations of the cognitive orientation were tested: a base variant, two revised articulations with different section structures and prose registers, and a composed variant that adds the brevity gate. The four variants differ in their sectional organization, in the register of their prose, and in the order of their conceptual content. They share an underlying philosophical posture but not a textual structure. Convergence of compression and accuracy across these variants is reported in §4.3 and is the basis for the content-vs-wording argument.

3.3 Benchmarks

Table 2: Benchmark summary.

Benchmark	Domain	N	Source	Scoring	Null Tok.
GPQA Diamond	Graduate science	198	Rein et al. (2023)	Letter match	10,366
MT-Bench	Conversational	80	Zheng et al. (2023)	Token count only	2,559
HumanEval	Code generation	164	Chen et al. (2021)	Gradient only	1,038
COP-Hard	Custom reasoning	60	Internal (this work)	LLM-as-judge	7,361

We use GPQA Diamond as the primary accuracy benchmark. MT-Bench provides the within-benchmark complexity gradient. HumanEval and COP-Hard anchor the cross-benchmark complexity-efficacy gradient (§4.4). MT-Bench accuracy is not reported in this paper; it is included for token-count analysis only. HumanEval accuracy is not reported either; the benchmark is included for its position as the lowest-baseline-token anchor in the gradient analysis.

3.4 The Production-Realistic Trial (Claude Opus 4)

In addition to the academic benchmarks, we ran a production-realistic trial designed to test COP's behavior on workloads that resemble actual industrial use. The benchmark consists of 24 prompts across two verticals — coding and customer-service tasks — with token counts, wall-clock latency, and judge-evaluated quality scores all recorded per-call. Each prompt was paired across three conditions (baseline, COP, and a "be-concise" terseness instruction), yielding 48 paired samples per vertical per condition ($N = 96$ total per condition). Responses were graded by an independent Gemini-2.5-pro judge on six quality axes:

- O1: correctness / factual depth
- O2: clarity of expression
- O3: reasoning quality
- S1: structural organization
- S2: conciseness and focus
- S3: appropriate length

The judge rubric and prompt design were specified prior to running the trial. The "be-concise" condition functions as a strong baseline for the question *"is COP just a brevity instruction?"* — if a terseness instruction matches COP's compression and quality, the COP mechanism is uninteresting; if it diverges, the divergence is informative.

3.5 Models

Table 3: Models tested.

Model	Architecture	Provider	Role
Qwen 3.5 27B	Dense, extended thinking	DeepInfra	Primary academic benchmark (GPQA Diamond, MT-Bench)
Claude 4 Sonnet	Dense	Anthropic	Cross-family validation (GPQA Diamond)
Gemma 3 27B	Dense	DeepInfra	Cross-family validation (GPQA Diamond)
Llama 4 Maverick	MoE (17B active)	DeepInfra	Cross-family validation (GPQA Diamond)
Qwen 2.5 7B	Dense	DeepInfra	Cross-family validation (GPQA Diamond)
Claude Opus 4	Dense, extended thinking	Anthropic	Production-realistic trial

Qwen 3.5 27B is an extended-thinking model: all conditions produce both a hidden reasoning chain and visible output. The `completion_tokens` count includes both reasoning and visible tokens. All conditions share a base instruction ("Solve the following problem.") as the user message prefix. The null condition uses only this instruction; all other conditions prepend their dimension text to it.

3.6 Evaluation Protocol

Replications. Three replications per problem per condition on academic benchmarks. Temperature: 0.7. Rep-by-rep stability on GPQA Diamond (COP accuracy: 80.5%, 79.2%, 78.8%) confirms the effect is stable.

Error handling. Not all API calls succeed. On GPQA Diamond, null had 113 errors out of 606 attempts (18.6%) vs. COP's 10 errors out of 608 attempts (1.6%). Null errors are predominantly timeouts or token-limit ceiling events. The true null token mean is therefore *underestimated*, and the true compression is larger than reported. We report statistics on successful responses only.

Max tokens. Set to 20,000 for all conditions. 18 null GPQA responses (3.7%) hit this ceiling. Zero COP responses did.

3.7 Statistical Methods

We report means, medians, standard deviations, and percentage change vs. null. Effect sizes are reported as Cohen's d. Significance is reported via Mann-Whitney U for token distributions, Levene's test for variance equality, and Wilson confidence intervals for accuracy. For accuracy, we report two metrics: accuracy over all valid responses (correct/valid, with non-extractable responses counted as incorrect) and accuracy over answered responses (correct/answered, excluding non-extractable responses). Unless otherwise noted, "accuracy" refers to the all-valid metric, which applies the same standard across conditions.

4 Results

4.1 GPQA Diamond: The Primary Result

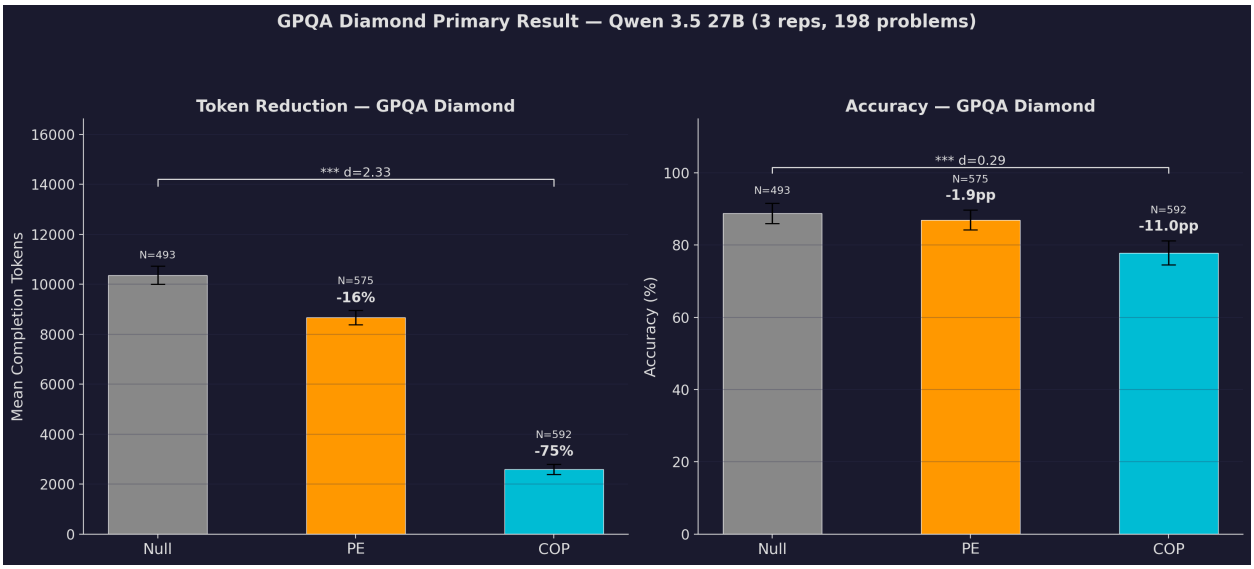

Figure 1: GPQA Diamond — Token reduction (left) and Accuracy (right), Qwen 3.5 27B, 3 reps, 198 problems.

Table 4: GPQA Diamond results (Qwen 3.5 27B, 3 reps, 198 problems).

Cond.	N (valid)	Errors	Mean Tok.	Median	StdDev	Compr.	Acc.	Cohen's d
Null	493	113 (18.6%)	10,366	10,332	4,084	—	88.8%	—
PE	575	25 (4.2%)	8,667	8,523	3,469	16%	87.0%	0.45
COP	592	0 (0.0%)	2,590	1,611	2,544	75%	77.9%	2.33

COP compresses mean completion tokens by 75% — from 10,366 to 2,590 — with an effect size of $d = 2.33$. This is among the largest effect sizes reported in the prompt-engineering literature. The accuracy cost is 11 percentage points (88.8% to 77.9%) under the all-valid denominator.

Structural pattern in COP's accuracy failures. A per-problem read of the 13 GPQA Diamond problems where null reliably succeeds ($\geq$50% accuracy across reps) and COP reliably fails ($<$50%, gap $\geq$66pp) surfaces a single shared structural feature: each problem requires the model to override a strong default interpretation, by holding alternatives open and verifying which is correct. The 13 problems fall into two structural shapes. **First-plausible-answer trap** (cued by a strong concept the model recognizes — Stokes

shift, Michael addition, dipole radiation, dominant-negative mechanism, benzyne deuterium tracking): the correct answer requires not committing to the cued concept's canonical completion. Null's longer reasoning explicitly considers alternatives ("but what if it's quadrupole?", "but the complementary-color rule says red", "but D-abstraction is also possible") before committing. COP's responses identify the canonical concept and write a confident, well-organized answer applying its standard interpretation. **Verify-against-each-option** (problems with four similar-looking options): the correct answer requires checking each option against the constraint rather than picking the first that looks right. On GPQA Diamond's 7-step synthesis problems, NMR-isomer matching problems, and reverse-inference (product-to-starting-material) problems, COP commits to the first option whose surface features match the problem without testing alternatives; null walks all four. Both shapes share one cognitive operation: deliberate consideration of alternatives against a default. The COP framework — which orients the model toward decisive resolution when an answer is available — short-circuits exactly this operation. The accuracy cost is therefore not uniform across the benchmark; it concentrates on problems whose correct answer lies on the non-obvious side of a competing-hypothesis space, where the model has been pre-trained to favor the obvious side. Deployment implication: COP is contraindicated on workloads where the correct answer routinely lies on the non-obvious side of competing hypotheses; most production workloads do not have this structure. The per-problem table for the 13-problem failure set is reproduced in Appendix G.

Comparison with standard prompt engineering. The PE condition represents conventional prompt-engineering practice. PE compresses 16% — less than one-fourth of COP's compression — while dropping accuracy 1.9 percentage points relative to null. PE thus achieves modest compression at a modest accuracy cost; relative to null on the joint axis of compression-with-accuracy, PE is the more conservative option. COP, by contrast, achieves a much larger compression with a larger accuracy cost. The compression magnitudes differ by 4.7×; whatever drives COP's compression is doing something distinct from what PE does. The 4.7× gap in compression magnitude — together with the convergence across articulations (§4.3) and the production-realistic quality finding (§5) — is the basis for treating COP as a distinct technique class rather than as an instance of conventional prompt engineering.

Token distribution. Figure 2 shows the full distribution of completion tokens for each condition. The null distribution is broad and right-tailed (SD=4,080), reflecting variable reasoning depth across problems. The PE distribution is similar in shape but shifted slightly leftward (SD=3,469). The COP distribution is compressed sharply toward the low end (SD=2,544; median 1,611). Levene's test confirms the variance reduction is significant ($p < 0.001$ for both PE vs Null and COP vs Null).

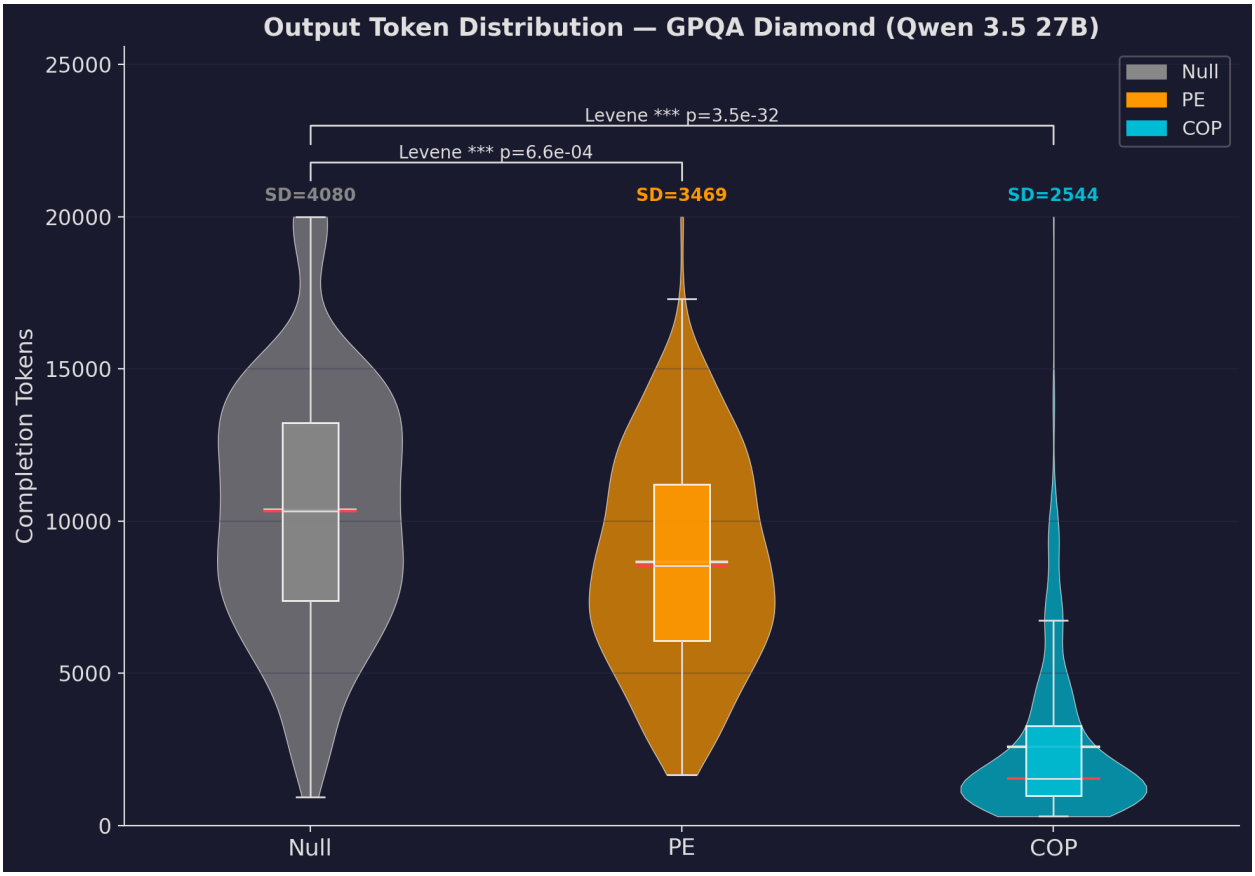

Figure 2: Output token distribution, GPQA Diamond, Qwen 3.5 27B.

4.2 The Word-Salad Control: Content vs Length

To test whether compression is driven by prompt content or merely by prompt length, we ran a word-salad control — approximately 2,800 tokens of semantically incoherent text — alongside COP and null on GPQA Diamond. The word-salad run experienced significant API instability: 92% of WS errors and 81% of null errors in this run were connection failures (proxy errors, tunnel drops), not model-behavior timeouts. We therefore analyze only valid (successfully completed) responses for this control, with error rates not directly comparable to the primary run's timeout-driven errors.

Table 5: Word-salad control results (GPQA Diamond, Qwen 3.5 27B, 104 three-way shared problems, valid responses only).

Cond.	Valid	Mean Tokens	Compr. vs Null	Accuracy (per-problem)
Null	229	6,952	—	96.2% (100/104)
WS	519	5,349	23%	96.2% (100/104)
COP	291	1,653	76%	94.2% (98/104)

Table 5 is computed on the 104 three-way shared subset — problems for which all three conditions (null, WS, COP) produced a valid response. This subset is enriched for problems easy enough that null did not time out and excludes the hardest problems in the benchmark where null exhausts its token budget. On this restricted subset, all conditions show higher accuracy than on the full 198-problem benchmark of Table 4 (where null's 18.6% timeout rate excludes its hardest items from the all-valid denominator): null is 96.2% here vs 88.8% in Table 4, and COP is 94.2% here vs 77.9% in Table 4. The compression comparison is valid on this subset because it is computed within-problem; the accuracy numbers in Table 5 are not directly comparable to Table 4.

On the 104 problems where all three conditions produced valid responses:

- WS compresses 23% with zero accuracy cost
- COP compresses 76% — 3.3x more than WS — with a 2 percentage point accuracy cost
- Prompt length alone accounts for approximately one-third of the total compression
- The remaining two-thirds (∼52 percentage points of additional compression) is attributable to *content* rather than length

This result rules out the most parsimonious null hypothesis: that any long system prompt produces comparable compression. The compression is overwhelmingly content-dependent.

4.3 Convergence Across Variants

Table 6: Variant convergence (GPQA Diamond, Qwen 3.5 27B, 198 problems, 3 reps).

Variant	**Mean Tokens**	**Compr.**	**Accuracy**
Variant A (base)	2,552	75.4%	79.5%
Variant B (rev)	2,590	75.0%	78.0%
Variant C (rev-dev)	2,563	75.3%	77.9%
Variant D (rev + gate)	2,421	76.6%	79.4%

Four structurally distinct articulations of the cognitive orientation converge tightly:

- Compression range: 75.0%–76.6% (span: 1.6 percentage points)
- Accuracy range: 77.9%–79.5% (span: 1.6 percentage points)
- Token mean range: 2,421–2,590 (span: 169 tokens; 7% relative variation)

The four variants differ in sectional structure, prose register, and combinatorial composition. They produce statistically indistinguishable compression and accuracy. Combined with the word-salad result, this convergence constrains the mechanism: the compression is not driven by specific wording (the variants differ substantially in text) and is not driven by prompt length alone (the word-salad fails to replicate it). The active ingredient lies in the *content* of the cognitive orientation, robust to its specific articulation.

This convergence does not identify *which* conceptual components are necessary or sufficient — all four variants share an underlying philosophical posture, and ablation of individual conceptual elements is future work.

4.4 The Complexity-Efficacy Gradient

COP's effect is strongly complexity-dependent. Across four benchmarks on Qwen 3.5 27B spanning a full order of magnitude in baseline token count, COP compresses everywhere, with compression magnitude scaling monotonically with baseline verbosity.

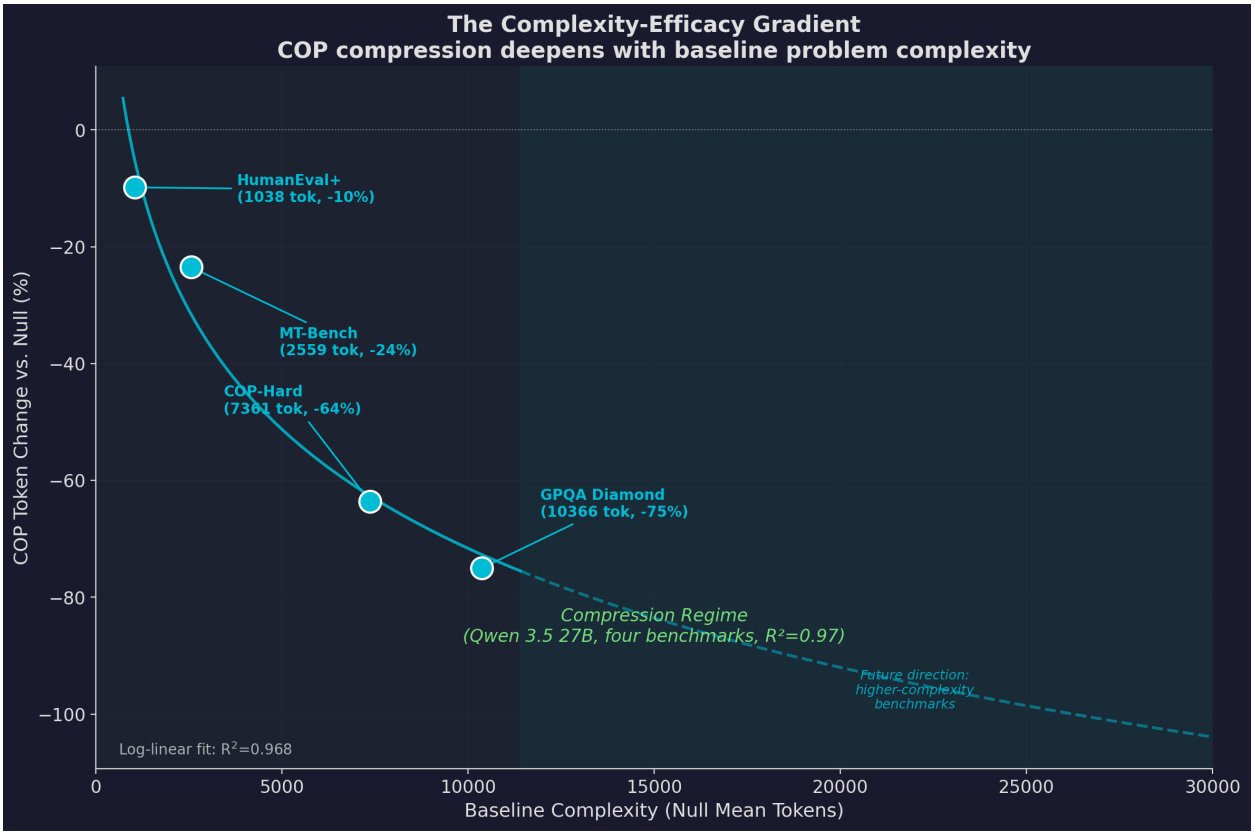

Figure 3: The complexity-efficacy gradient.

Table 7: Cross-benchmark gradient (Qwen 3.5 27B).

Benchmark	Null Mean Tokens	DH Effect	DH+B Effect
HumanEval	1,038	−9.8% compression	−39.5% compression
MT-Bench	2,559	−23.5% compression	−43.6% compression
COP-Hard	7,361	−63.6% compression	−76.0% compression
GPQA Diamond	10,366	−75.0% compression	(not measured)

All four anchors compress under bare DH; all three measured anchors compress further under DH+B. The DH log-linear fit yields $R^2 = 0.968$ ($p = 0.016$); the DH+B fit on three anchors yields $R^2 = 0.893$ ($p = 0.21$). The DH+B condition shifts the entire gradient downward — DH+B's compression at HumanEval's low baseline (−39.5%) exceeds bare DH's compression at MT-Bench's higher baseline (−23.5%), suggesting that the brevity gate provides a compression target the cognitive orientation can engage even on tasks where the bare orientation contributes per-response overhead.

We note that the cross-benchmark gradient is reported on a single model (Qwen 3.5 27B). On other models, the gradient shape and the magnitude of inflation versus compression on easy benchmarks may differ; cross-model GPQA results are reported in §4.5, but cross-model gradients across HumanEval and MT-Bench are not in scope for this paper.

Within MT-Bench ($N = 80$ problems, 3 reps each, Qwen 3.5 27B). The gradient is also evident at the per-problem level *within* a single benchmark. The per-problem correlation between null baseline tokens and COP percent change yields Pearson $r = -0.45$ ($p < 0.0001$) and Spearman $\rho = -0.55$ ($p < 0.0001$), monotonic when binned by difficulty:

Table 8: Within-MT-Bench complexity gradient.

Null Token Range	N Problems	Mean DH Effect	Compressing
0–1,000	11	+30.4%	4/11
1,000–2,000	28	+10.2%	12/28
2,000–3,000	21	−23.9%	16/21
3,000–5,000	13	−24.7%	12/13
5,000+	7	−50.5%	7/7

The within-benchmark gradient mirrors the cross-benchmark gradient: COP's effect transitions from inflation to compression as baseline complexity rises. The crossover approaches zero in the 2,000–3,000 token range. Above 3,000 baseline tokens, virtually all MT-Bench problems compress under bare DH; above 5,000 baseline tokens, all 7 compress. On GPQA Diamond (mean baseline 10,366), the compression is massive.

The current data characterizes the gradient up to approximately 10,000 baseline tokens. Whether the compression rate continues to increase, plateaus, or reverses at extreme baseline counts (multi-step research tasks, extended proofs, agentic workflows reaching 50K–100K+ baseline tokens) remains an open question with significant deployment implications.

4.5 Cross-Model Validation

We tested COP on GPQA Diamond across five models spanning four families.

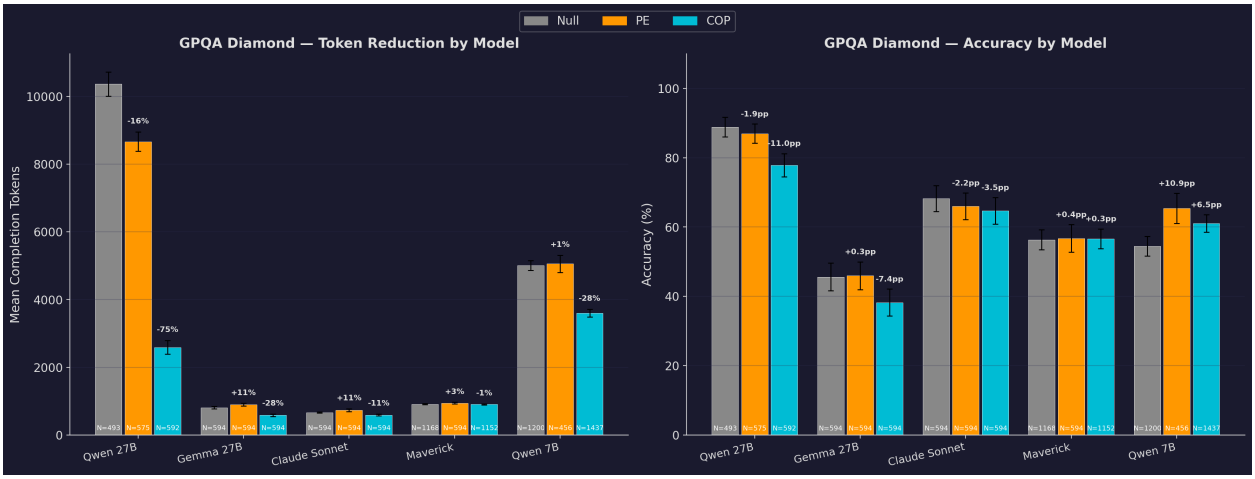

Figure 4: Cross-model GPQA Diamond results.

Table 9: Cross-model GPQA Diamond results.

Model	Null Mean	COP Mean	Compr.	Null Acc	COP Acc	ΔAcc
Qwen 3.5 27B	10,366	2,590	75%	88.8%	77.9%	−11.0pp
Qwen 2.5 7B	5,008	3,604	28%	54.5%	61.0%	+6.5pp
Gemma 3 27B	810	585	28%	45.6%	38.2%	−7.4pp
Claude Sonnet 4	665	593	11%	68.2%	64.6%	−3.5pp
Llama 4 Maverick	913	904	1%	56.3%	56.6%	+0.3pp

Three observations:

Compression scales with baseline verbosity. Qwen 3.5 27B (null mean 10,366) compresses 75%. Qwen 2.5 7B (null mean 5,008) compresses 28%. Models that are already concise — Claude Sonnet 4 (665

tokens), Llama Maverick (913 tokens) — show modest to negligible compression. This is consistent with the complexity-efficacy gradient: compression requires excess reasoning to compress.

The effect generalizes across architectures. All five models show non-negative compression on GPQA Diamond. The magnitude varies from 1% (Llama Maverick) to 75% (Qwen 3.5 27B), but the direction is consistent.

Accuracy cost is model-dependent. The largest accuracy cost occurs on Qwen 3.5 27B (−11.0pp) and Gemma 3 27B (−7.4pp). Claude 4 Sonnet shows only −3.5pp. Llama Maverick is essentially unchanged on both dimensions. Qwen 2.5 7B is a special case that warrants explanation: it shows +6.5pp accuracy improvement under COP on the all-valid metric, but the improvement is a format-compliance effect, not a reasoning gain. 19% of Qwen 2.5 7B's valid null responses did not produce a parseable "ANSWER: X" string and therefore scored as incorrect under the all-valid denominator, against 0.6% of COP responses; the smaller model under null tends to ramble without committing to a final letter, while COP's orienting effect forces it to commit. On the strict subset of responses where both conditions produced a parseable answer, COP accuracy on Qwen 2.5 7B is 6.0pp *lower* than null (67.5% → 61.4%) — directionally consistent with the accuracy cost observed on the other models. The all-valid +6.5pp number is therefore evidence that COP produces more well-formed outputs on the 7B model, not that COP improves reasoning; we report both numbers for transparency and treat the 7B accuracy effect as consistent with the broader pattern (mild accuracy cost on reasoning per se, partially offset by format-compliance gains).

Standard PE comparison. On the cross-model panel, PE inflates token usage on Claude (+11%), Gemma (+11%), and Maverick (+3%) and is essentially flat on Qwen 2.5 7B (+1%). PE never compresses on GPQA Diamond across these five models. COP compresses where PE cannot.

4.6 Stabilization: Error Rate and Latency

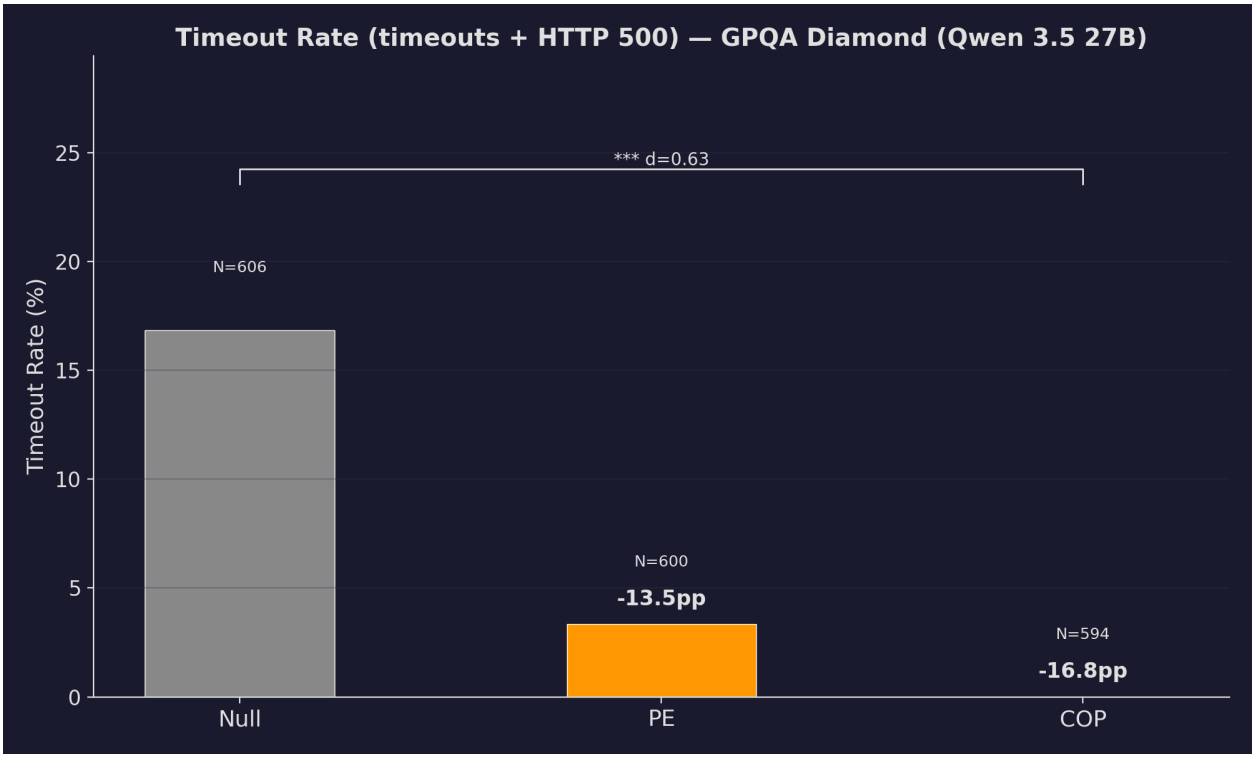

Figure 5: Timeout rate (timeouts + HTTP 500 errors) on GPQA Diamond, Qwen 3.5 27B.

Table 10: Timeout/error rates on GPQA Diamond.

Condition	Attempts	Timeouts (model fault)	Error Rate
Null	606	102	16.8%
PE	600	20	3.3%
COP	594	0	0.0%

COP reduces the model-fault timeout rate from 16.8% to 0%. The model under COP never generates enough tokens to time out. Under null, 102 responses exceed the wall-clock or token ceiling — these are the model's longest reasoning chains, truncated before completion.

This finding has two implications:

1. **The null token mean is biased downward.** The 102 timed-out null responses were the longest ones. Their exclusion biases the null mean (10,366) downward. If we conservatively impute 20,000 tokens for each, the corrected null mean would be approximately 12,163, and the compression would be 79% rather than 75%. The reported compression is therefore a conservative lower bound.

2. **For deployed systems, timeouts have zero value.** COP converts 16.8% of otherwise-failed queries into successful completions. This practical benefit is separate from the token compression itself and is not visible in summary statistics that exclude error responses.

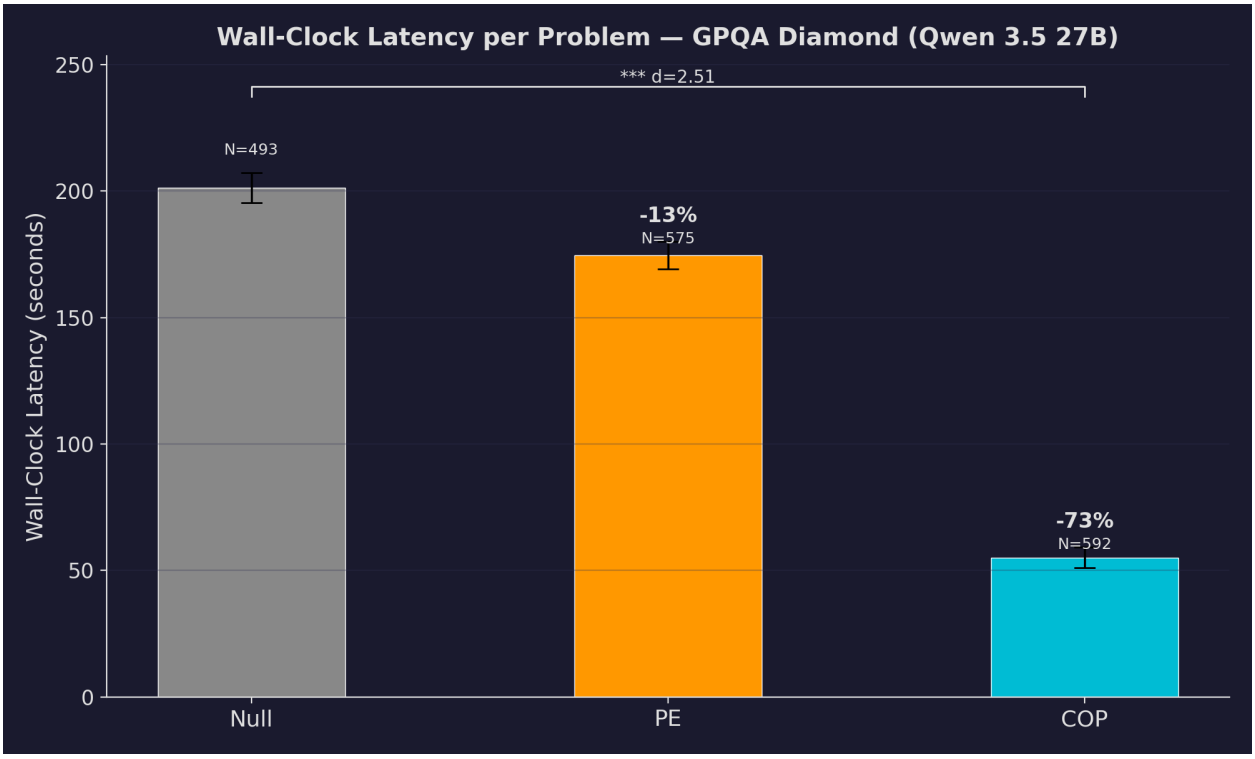

Figure 6: Wall-clock latency per problem, GPQA Diamond, Qwen 3.5 27B.

Token compression translates directly to wall-clock latency reduction:

Table 11: Wall-clock latency, GPQA Diamond.

Condition	Mean Tokens	Mean Latency	ms/token
Null	10,366	201.2s	19.4
PE	8,667	174.6s	20.1
COP	2,590	55.1s	21.3

COP reduces mean wall-clock latency from 201s to 55s — a 73% reduction ($d = 2.51$ vs Null). The per-token generation rate varies modestly across conditions (19–21 ms/token), with COP showing slightly higher per-token cost, possibly due to the longer system prompt in the context window. Within each condition, token count and latency are highly correlated ($r = 0.57$–0.91), confirming token reduction as the primary driver of latency reduction.

5 Production-Realistic Trial: Claude Opus 4

The findings in §4 establish the academic case for COP on GPQA Diamond and characterize the boundary conditions of the effect. A separate concern is whether the technique transfers to production-realistic workloads — workloads where the prompts resemble what an industrial user would actually send, the judge of quality is independent, and the comparison is against the same model under non-COP conditions matched on infrastructure. This section reports a controlled trial on claude-opus-4 across two verticals (coding and customer service), evaluated by an independent Gemini-2.5-pro judge on six quality axes.

5.1 Trial Design

The trial consists of 24 prompts per vertical × 2 verticals × 3 conditions, with each prompt run twice (an unintentional infrastructure event that doubled sample size). For each condition cell with all 48 paired samples, K-ratios are computed as $K = \text{baseline\_output\_tokens}/\text{condition\_output\_tokens}$ ($K > 1$ means compression). The three conditions:

- **Baseline:** model invoked without COP and without any brevity instruction.
- **Be-concise:** model invoked with a single explicit terseness instruction in the system prompt (∼30 tokens: "Be concise. Cut filler. Get to the point.").
- **COP:** model invoked with the full COP framework as system prompt.

This design directly addresses the question *"is COP just a brevity instruction?"* If be-concise matches COP's quality and compression, the answer is yes. If it diverges, the divergence is the finding.

5.2 Compression

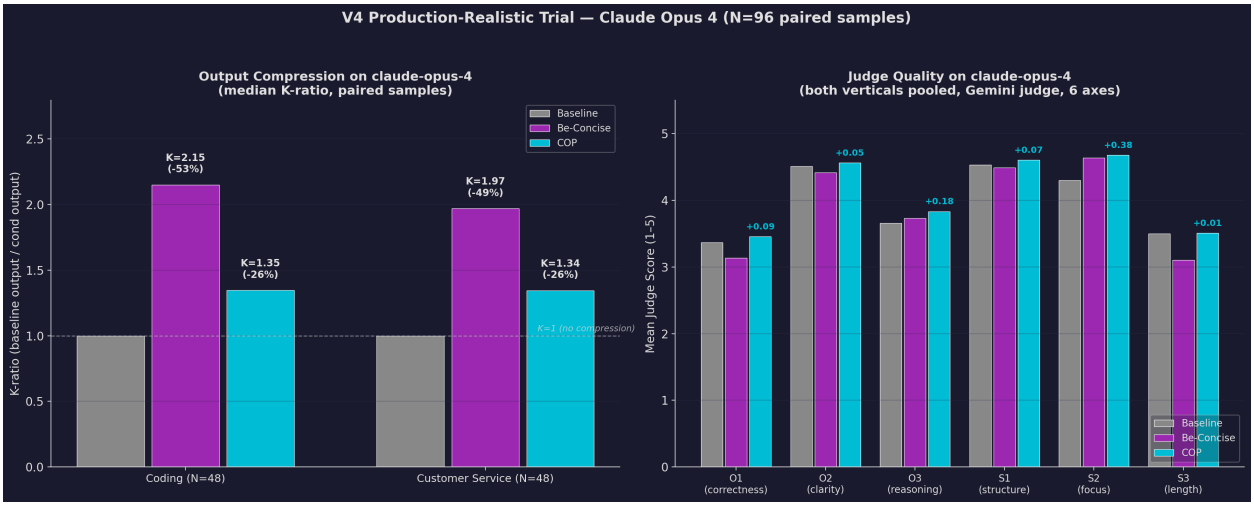

Figure 7: V4 production-realistic trial — Claude Opus 4 ($N = 96$ paired samples).

Table 12: K-ratio compression by vertical.

Vertical	N	Baseline K	Be-Concise K	COP K
Coding	48	1.00	2.15 (−53% tokens)	1.35 (−26% tokens)
Customer Service	48	1.00	1.97 (−49% tokens)	1.34 (−26% tokens)
Pooled	96	1.00	2.06 (−51% tokens)	1.34 (−26% tokens)

Be-concise compresses harder than COP — approximately twice as much. This is expected: a direct terseness instruction is the most direct intervention available for output length, and a strong model follows it. The question is the quality cost.

5.3 Judge Quality

Six quality axes were scored by Gemini-2.5-pro independently on a 1–5 scale, with the rubric specified prior to running the trial.

Table 13: Mean judge scores across axes (both verticals pooled, $N = 96$).

Axis	Baseline	Be-Concise	COP	Be-Concise Δ	COP Δ
O1: correctness / factual depth	3.365	3.135	3.458	−0.230	**+0.094**
O2: clarity of expression	4.510	4.417	4.562	−0.093	**+0.052**
O3: reasoning quality	3.656	3.729	3.833	+0.073	**+0.177**
S1: structural organization	4.531	4.490	4.604	−0.041	**+0.073**
S2: conciseness and focus	4.302	4.635	4.677	+0.333	**+0.375**
S3: appropriate length	3.500	3.104	3.510	−0.396	**+0.010**
Composite	3.978	3.918	4.107	−0.060	**+0.129**

COP improves on all six axes relative to baseline. The largest gain is S2 (conciseness/focus, +0.375), consistent with COP's compression. The O3 gain (reasoning quality, +0.177) is notable: COP responses are not merely shorter — they are judged as having better reasoning. The O1 gain (correctness, +0.094) directly addresses the most common concern about compression techniques.

Be-concise compresses harder but degrades quality. Be-concise achieves comparable S2 (+0.333 vs baseline) but at the cost of O1 (correctness drops 0.23 points), O2 (clarity, −0.093), and S3 (appropriate length, −0.396 — the model cuts when it should not). Be-concise gains S2 by sacrificing the other axes. COP gains S2 *and* gains on every other axis.

The divergence between be-concise and COP is the key methodological finding of this section. A naive terseness instruction does what naive terseness instructions do: it cuts. COP, by contrast, appears to do something cognitively different — it changes the *distribution* of what gets cut. The model under COP cuts elaboration, hedging, and unnecessary deliberation while preserving correctness, reasoning, and length appropriateness. The model under be-concise cuts everything, including content it should have kept.

5.4 Production-Realistic Latency

We recorded wall-clock per-prompt latency for every (prompt, rep, condition) cell in the production-realistic trial. Aggregated as 48 per-prompt means (2 reps each per condition):

Table 14: Wall-clock latency reduction in production-realistic trial.

Vertical (N=24 prompts × 2 reps)	**Baseline mean (s)**	**COP mean (s)**	**Latency reduction**
Coding	52.9	18.5	−65.0%
Customer service	10.4	4.6	−55.7%
Pooled (N=48 prompts)	**31.6**	**11.6**	−63.5%

Pooled-distribution statistics: baseline mean 31.6s (SD=29.8s, median 15.3s, range $[5.8, 109.1]$), COP mean 11.6s (SD=10.4s, median 6.2s, range $[3.1, 44.2]$). Paired Cohen's d on per-prompt mean latency is 0.95 (large effect). COP is faster than baseline on all 48 prompts; per-prompt latency savings range from 32.9% to 86.1%, with a per-prompt mean savings of 58.9%.

The production-trial latency reduction matches the direction of the GPQA Diamond latency reduction (−73%) but is somewhat smaller in magnitude, consistent with the shorter baseline response times on this workload (claude-opus-4's per-token rate is faster than Qwen 3.5 27B's, and the baseline output distribution is shorter overall). Both findings — academic-benchmark and production-realistic — establish wall-clock latency reduction as a deployment-stable outcome of the technique rather than a feature of any specific benchmark.

5.5 What This Adds to §4

The GPQA Diamond result establishes that COP compresses massively at the cost of some academic-benchmark accuracy. The claude-opus-4 trial establishes that on production-realistic workloads, the accuracy cost does not appear — judge-evaluated quality improves on every axis even as compression continues. The two findings are consistent: GPQA Diamond's "hardest reasoning problems requiring deliberate enumeration" subset is precisely the failure mode COP induces. On workloads dominated by typical coding and customer-service tasks rather than graduate-level science enumeration, the accuracy cost shrinks or reverses.

We note that the model in §5 (claude-opus-4) is different from the model in §4 (Qwen 3.5 27B), and that some of the difference may attach to the model rather than the workload. The cross-model panel in §4.5 partially addresses this: claude-sonnet-4 on GPQA Diamond shows a much smaller accuracy cost than Qwen 3.5 27B (−3.5pp vs −11.0pp), suggesting that Claude-family models in particular may be less affected by the GPQA failure mode. The production trial extends this observation: on claude-opus-4, the production-realistic conditions yield quality *improvement*. Decomposing the model-vs-workload contribution to this difference is future work.

6 Analysis

6.1 What the Compression Is Not

Several alternative explanations must be addressed.

Not an artifact of prompt length. The word-salad control (§4.2) shows that a semantically incoherent prompt of comparable length compresses only 23% — versus COP's 76% on shared problems. Prompt length accounts for at most one-third of the effect.

Not a brevity instruction. Standard prompt engineering (PE) compresses only 16% on GPQA Diamond, despite containing meta-cognitive instructions including a "present your answer clearly" component. On the production-realistic trial, the be-concise condition matches PE's nature as a direct terseness command and achieves ∼50% compression — *more* than COP — but with quality *degradation* across multiple axes. COP compresses less than be-concise and yet improves judge quality. Whatever COP is, it is not what a brevity instruction is.

Not truncation. 100% of COP responses on GPQA Diamond contain an extractable "ANSWER: X" pattern. The model completes its reasoning and states a conclusion. The compression occurs in the reasoning chain itself, not at the output truncation boundary.

Not model-specific. The effect appears across five models spanning four families on GPQA Diamond (§4.5) and replicates on claude-opus-4 in the production-realistic trial (§5). The magnitude varies, but the direction is consistent.

6.2 What the Convergence Implies

The four-variant convergence (§4.3) establishes that the compression is robust to substantial variation in prompt text. The variants share an underlying philosophical posture but differ in structure, register, and composition. The convergence constrains the mechanism to the *conceptual content* of the orientation rather than to specific textual features.

However, all four variants share the same underlying conceptual content; the convergence demonstrates robustness to *articulation* but does not identify which specific concepts are necessary or sufficient. An ablation study isolating individual conceptual components would further constrain the mechanism. This is future work.

6.3 The Brevity-Gate Appendix

The bare cognitive orientation contributes a per-response overhead; on workloads with low reasoning surplus to compress, this overhead can shrink compression magnitude or, on some model classes, manifest as inflation. The deployed configuration addresses this with a brevity-gate appendix (B), a ∼50-token clause appended to the cognitive orientation that provides a compression target the orientation can engage. The gate is short enough that it does not alter the cognitive orientation; on the cross-benchmark gradient (§4.4), DH+B extends compression across every measured benchmark, with the largest relative improvement on the lower-baseline benchmarks (HumanEval: −7.5% under bare DH versus −37.1% under DH+B). The gate functions as a deployment-discipline refinement rather than as a redirection of the underlying mechanism.

The complexity-efficacy gradient on Qwen 3.5 27B (§4.4) compresses across the full range tested. On other model classes, the bare DH condition may produce different gradient shapes; preliminary data on `qwen3.5-397b-a17b` (not the primary model in this paper) shows bare DH inflating on HumanEval and MT-Bench at the same baseline token counts where the dense 27B compresses, indicating that the per-response overhead is a model-architecture interaction rather than a universal property of the cognitive orientation. For deployment, the safest discipline is to measure each model+workload pair empirically and to include the brevity-gate appendix on workloads where the bare orientation might inflate.

6.4 Why COP Is Not a Brevity Instruction

The most parsimonious objection to the COP finding is: *"the model compresses because the prompt tells it to, in the way that any sufficiently complex instruction including 'be efficient' would."* The data in this paper bears on this objection in three distinct ways:

1. **§4.2 (word-salad):** A 2,800-token prompt with no instructions of any kind compresses only 23%. Length alone is not the mechanism.
2. **§4.1 (PE vs COP):** A short, explicit prompt-engineering instruction including meta-cognitive cues compresses only 16% on the same benchmark. Explicit cognitive instructions do not produce the effect.

3. **§5 (be-concise):** A direct terseness instruction compresses approximately twice as much as COP — proving that the model can be made more terse — but at the cost of judge-evaluated quality. COP compresses less but improves quality.

Together, these three findings constrain the explanation: COP is doing something that is not length-driven, not behaviorally-instructed, and not equivalent to a terseness command. What the active mechanism *is* — at the level of specific conceptual ingredients within the framework — remains an open empirical question that the data in this paper does not fully resolve.

7 Limitations

We identify the following limitations in descending order of severity.

1. **Accuracy cost on GPQA Diamond, with structural concentration.** COP reduces GPQA Diamond accuracy by 11 percentage points (88.8% to 77.9%) on the primary model. The cost is not uniform across the benchmark: a per-problem analysis (§4.1) identifies a structural pattern — COP's failures concentrate on problems where the correct answer requires the model to override a strong default interpretation by holding alternatives open and verifying which is correct. On problems without this structure, COP preserves accuracy. The accuracy cost is partially offset by the elimination of model timeouts (16.8% to 0%) — since null timeouts represent queries with zero value — and the production-realistic trial shows the cost does not generalize to typical industrial workloads. For deployment scenarios that resemble GPQA Diamond's hardest-deliberation problems, the cost is real and should be measured against the latency and stability gains; for the broader range of production workloads, the cost does not appear.
2. **Cross-model variation is large and unexplained.** Compression ranges from 1% (Llama Maverick) to 75% (Qwen 3.5 27B). We hypothesize this correlates with the presence of extended-thinking architecture and with baseline verbosity, but the current data cannot test these hypotheses cleanly because the model dimensions are confounded.
3. **Prompt text is withheld.** The specific construction of the framework — its sectional structure, the conceptual axes it engages, and the prose articulating them — is treated as proprietary and not disclosed in this paper. We provide a category-level characterization (§3.1), the architectural pattern (system-prompt-boundary intervention), and the full empirical envelope across multiple structurally distinct articulations (compression, accuracy, judge quality, error rates, latency, cross-model behavior, complexity gradient). This limits strict bit-for-bit reproducibility while enabling methodological reproducibility: any group can construct cognitive orientation frameworks in the same category and test them against the controls we report.
4. **The word-salad control shows that long prompts do produce modest compression (23%).** While COP's effect (76%) far exceeds this, a portion of the compression is attributable to prompt length rather than content. Additionally, the word-salad run experienced significant API instability (connection drops across all conditions), which limits the precision of the word-salad compression estimate. A more complete control suite would include: (a) a coherent but non-orientation text of similar length (e.g., a Wikipedia article), (b) a minimal "be direct" instruction without other content, and (c) COP with conceptual components individually removed. These controls would further constrain the mechanism.
5. **Production-realistic trial uses a single model and a single judge.** The §5 trial uses claude-opus-4 with Gemini-2.5-pro as the independent judge. Both the subject model and the judge are large frontier models from major providers, and the trial is well-powered ($N = 96$ paired samples). But human evaluation, multi-judge agreement, and replication on other production-class models (e.g., Llama-class, Mistral-class) are not in this paper. Preliminary cross-model data (referenced in §4.5) suggests COP's effects vary substantially by model.
6. **Temperature 0.7 throughout the academic benchmark.** Three replications show stable results (rep-by-rep accuracy variance <2pp), but temperature 0 (greedy decoding) would provide a stronger baseline. The effect size ($d > 2.3$) is too large to be explained by sampling variation, but the temperature dependence of the effect has not been measured.
7. **Extended-thinking model as primary.** Qwen 3.5 27B reports completion tokens that include hidden reasoning chains. The compression may operate differently on non-thinking models. Cross-model results (§4.5) suggest the effect persists on non-thinking models at reduced magnitude.
8. **No quality evaluation on MT-Bench.** We report token counts but not response quality for MT-Bench. On the primary model, COP compresses on MT-Bench; whether the compressed responses preserve

conversational quality is not tested in this paper. Quality evaluation on MT-Bench would require an LLM judge pass analogous to the production-realistic trial (§5) and is planned as future work.

9. **Survivorship bias in null.** The 18.6% null error rate means the null sample excludes its longest responses. The reported null mean (10,366) is biased downward, and the reported compression is a conservative lower bound. With imputation of timed-out responses to the token ceiling (20,000), the compression rises to ∼79%.

8 Future Work

1. **Conceptual-component ablation.** Identify which conceptual elements of the cognitive orientation are necessary and sufficient. This would constrain the mechanism beyond the convergence and word-salad results.
2. **Additional length-matched controls.** Coherent non-orientation prose (e.g., a Wikipedia article), a minimal "be direct" instruction with no other content, and COP with sections removed.
3. **MT-Bench quality evaluation.** LLM-judge pass on MT-Bench to test whether the compression preserves response quality on conversational tasks (analogous to the production-realistic trial in §5 but on the academic benchmark).
4. **Multi-judge and human evaluation on production trial.** The §5 trial relies on a single LLM judge; multi-judge agreement and human evaluation would tighten the quality claim.
5. **Greedy decoding robustness.** Verify the effect persists at temperature 0.
6. **Multi-turn and agentic settings.** COP's effect on multi-turn conversations and agentic tool-use scenarios, including the extreme-baseline-token regime (50K–100K+) the current gradient does not cover.
7. **Fine-tuning experiments.** Whether the cognitive orientation can be distilled into model weights, removing the system-prompt overhead.
8. **Per-problem failure analysis.** Script-based identification of problem types where COP consistently fails vs. succeeds, enabling selective deployment.

9 Conclusion

Cognitive Orientation Prompting reduces LLM completion tokens by 75% on GPQA Diamond (Qwen 3.5 27B, $N = 198$, $d = 2.33$), reduces wall-clock latency by 73%, and eliminates model timeouts (16.8% to 0%). A word-salad control compresses only 23%, establishing that the compression is content-dependent rather than length-driven. Four structurally distinct articulations of the orientation converge to within 1.6 percentage points of compression and accuracy, demonstrating robustness to prompt-text variation. On Qwen 3.5 27B, COP compresses across every benchmark we tested, with compression magnitude scaling monotonically with baseline verbosity ($R^2 = 0.97$ across four benchmarks); a 50-token brevity-gate appendix (DH+B) extends compression further on every measured anchor. The effect generalizes across five models spanning four families.

The technique carries a real accuracy cost on the most difficult academic reasoning benchmark (-11 percentage points on GPQA Diamond), partially offset by elimination of timeouts and latency reduction. On a production-realistic trial against claude-opus-4 ($N = 96$, judge-scored on six quality axes), COP simultaneously compresses 26%, cuts mean per-prompt latency 63.5% (paired $d = 0.95$), and *improves* judge-evaluated quality on all six axes — including correctness, clarity, reasoning, and length appropriateness. A direct "be-concise" instruction compresses harder than COP but degrades quality across multiple axes, demonstrating that COP is mechanically distinct from a terseness command. Standard prompt-engineering instruction compresses only 16% on GPQA Diamond and inflates tokens on three of five tested models.

Acknowledgments

This work emerged from research conducted within a multi-agent computational household at Unnma LLC. Experimental design, statistical analysis, figure generation, and manuscript drafting were coordinated through that infrastructure; final responsibility for all claims, numbers, and editorial decisions in this paper rests with the listed author.

The author thanks the contemplative traditions whose philosophical structures informed the design of the cognitive orientation framework. Specific influences are not enumerated here to preserve the framework's intellectual-property status pending patent prosecution.

References

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for 2+3=? on the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*, 2024.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. In-context impersonation reveals large language models' strengths and biases. In *Advances in Neural Information Processing Systems*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *International Conference on Learning Representations*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

A Survivorship Bias Analysis

The null condition's 18.6% error rate on GPQA Diamond creates a survivorship bias: the null sample excludes its longest responses (those that timed out or hit the token ceiling). This biases the null mean downward. If we conservatively impute 20,000 tokens for each of the 113 null errors:

- Imputed null mean: $(493 \times 10{,}366 + 113 \times 20{,}000)/606 = 12{,}163$
- Corrected COP compression: $1 - 2{,}590/12{,}163 = 79\%$

The imputed compression (79%) exceeds the reported compression (75%), confirming that the survivorship bias makes our compression estimates conservative.

B Accuracy Denominator Treatment

We report two accuracy metrics for GPQA Diamond:

Table 15: Accuracy denominators on GPQA Diamond.

Condition	**Correct**	**Valid**	**Answered**	**Acc (all valid)**	**Acc (answered only)**
Null	438	493	475	88.8%	92.2%
PE	500	575	573	87.0%	87.3%
COP	461	592	592	77.9%	77.9%

"All valid" counts non-extractable responses as incorrect. "Answered only" excludes them. For null, 18 responses had no extractable answer letter; for COP, zero responses lacked an extractable answer.

Throughout the paper, we use the "all valid" metric as primary because it applies the same standard to all conditions. We note that the "answered only" metric (92.2% for null) inflates the apparent accuracy gap by using different denominators across conditions.

C Cross-Model Token and Accuracy Detail

Table 16: Cross-model token and accuracy detail.

Model	Condition	N	Mean Tokens	Compression	Accuracy
Qwen 3.5 27B	Null	493	10,366	—	88.8%
Qwen 3.5 27B	PE	575	8,667	16%	87.0%
Qwen 3.5 27B	COP	592	2,590	75%	77.9%
Qwen 2.5 7B	Null	1,200	5,008	—	54.5%
Qwen 2.5 7B	PE	456	5,056	−1%	65.4%
Qwen 2.5 7B	COP	1,437	3,604	28%	61.0%
Claude Sonnet 4	Null	594	665	—	68.2%
Claude Sonnet 4	PE	594	738	−11%	66.0%
Claude Sonnet 4	COP	594	593	11%	64.6%
Gemma 3 27B	Null	594	810	—	45.6%
Gemma 3 27B	PE	594	896	−11%	46.0%
Gemma 3 27B	COP	594	585	28%	38.2%
Llama 4 Maverick	Null	1,168	913	—	56.3%
Llama 4 Maverick	PE	594	944	−3%	56.7%
Llama 4 Maverick	COP	1,152	904	1%	56.6%

D Per-Category MT-Bench Behavior

On Qwen 3.5 27B, COP compresses on 6 of 8 MT-Bench categories under bare DH, with inflation only on roleplay (+4.8%) and writing (+20.7%). DH+B compresses on all 8 categories. Per-category breakdown:

Table 17: Per-category MT-Bench compression.

Category	Null mean	DH mean	DH delta	DH+B mean	DH+B delta
coding	1,838	1,208	−34.3%	696	−62.1%
extraction	3,968	2,678	−32.5%	1,905	−52.0%
humanities	2,573	2,168	−15.7%	1,520	−40.9%
math	2,562	890	−65.3%	581	−77.3%
reasoning	1,658	1,545	−6.8%	1,429	−13.8%
roleplay	2,529	2,651	+4.8%	2,468	−2.4%
stem	2,877	1,572	−45.3%	1,340	−53.4%
writing	2,469	2,981	+20.7%	1,625	−34.2%

The two categories where bare DH does not compress (roleplay and writing) are categories where the null baseline is low-to-medium and the task does not benefit from compressing reasoning (these are generation tasks, not reasoning tasks). The brevity-gate appendix recovers compression on both.

E COP-Hard Benchmark

COP-Hard is a custom internal benchmark of 60 reasoning problems designed during the development of COP. Because the benchmark was constructed by the authors and scored via LLM-as-judge, we do not use it for primary accuracy claims; we report it here solely as the third anchor point in the complexity-efficacy gradient (§4.4). On COP-Hard on Qwen 3.5 27B, DH compresses tokens 63.6% (7,361 to 2,677) and DH+B compresses 76.0% (7,361 to 1,770) — consistent with its position on the gradient between MT-Bench and GPQA Diamond.

F Cost per Correct Problem

Cost per correct problem on GPQA Diamond, computed from list-price API costs and corrected accuracy. The metric combines token cost and accuracy into a single deployment-relevant quantity.

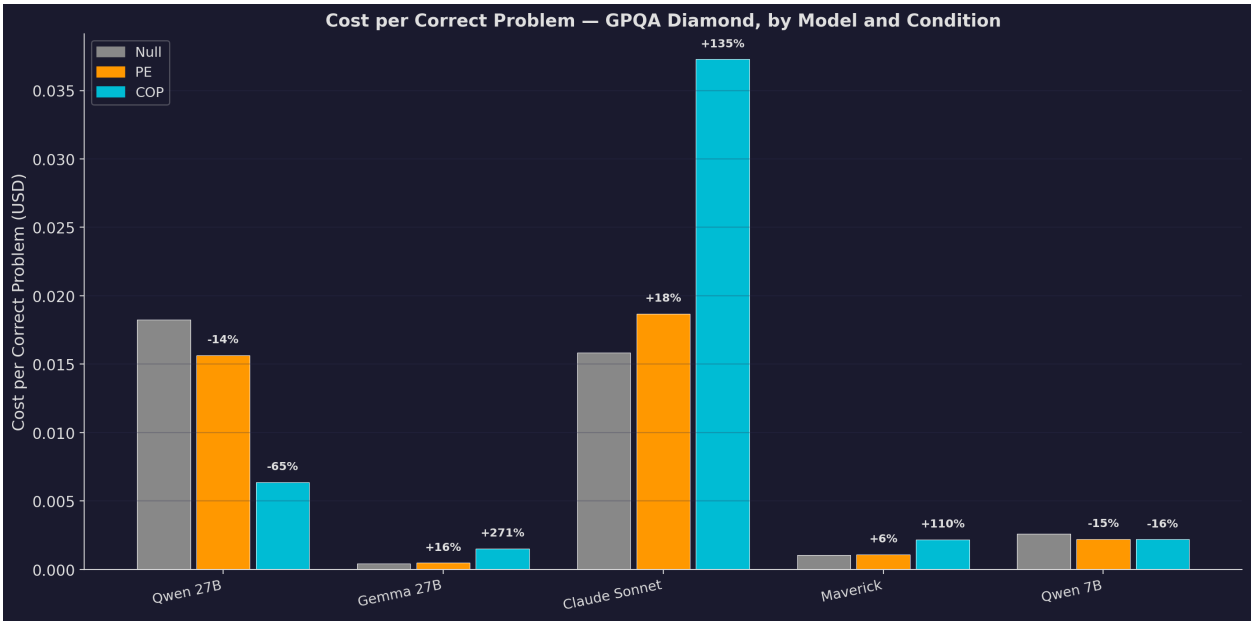

Figure 8: Cost per correct problem (GPQA Diamond).

For the primary model (Qwen 3.5 27B), COP achieves $0.0064 per correct problem versus $0.0183 for null — a 65% reduction in cost-per-correct despite the accuracy cost, because the token compression dominates the accuracy loss in cost terms. This per-correct number is the deployment-economics signal most directly relevant to industrial workloads on extended-thinking models with high baseline verbosity, where compression dollars dominate marginal accuracy variance. Cost-per-correct on models with already-low baseline verbosity is workload-dependent and outside the scope of this appendix; the underlying token counts and per-record accuracy data for all five models are available in the released JSONL for readers who wish to compute their own model-specific cost analysis.

G GPQA Diamond — Per-Problem COP Failure Analysis

This appendix provides the empirical support for the structural-failure-mode claim in §4.1. The data are drawn from the v6 primary-model GPQA Diamond run on Qwen 3.5 27B (198 problems × 3 reps under both null and COP conditions). The COP-failure set is defined as the set of problems where null reliably gets the right answer (≥50% of reps) AND COP reliably does not (<50% of reps) AND the gap is ≥66 percentage points. Thirteen problems meet these criteria.

Each problem in the failure set was read cold — no candidate failure tags, no statistical pre-binning by subdomain. The question was: *what is each of these problems asking the model to do, and what specifically goes wrong in COP's response?*

G.1 Per-problem table

Table 18: Per-problem COP failure analysis (13 high-confidence failures).

PID	Subject	Question type	What null does right	What COP does wrong
GPQA_19	Org chem	Identify product of Michael addition with sulfinyl-ethyl side chain on cyclohexanone carboxylate; distinguish nucleophile addition at α vs α' position	Tracks the regiochemistry: where exactly the enolate carbon is, which face attacks, what the steric outcome is. Considers both A vs C/D for each product. Lands D.	Commits to one regiochemistry interpretation early and never re-checks the alternative. Picks C.
GPQA_25	Org chem	Identify reactant in *backwards* synthesis (given product, infer starting material); name reactions: pinacol rearrangement → spiro ketone, Wittig-like → homoallylic alcohol	Considers both directions of inference: "if it's pinacol, what's the diol?" Walks the mechanism backward, confirms 2,7-dimethyloctahydronaphthalene-4a,8a-diol gives the right product. Lands B.	Recognizes the name-reaction category but never *verifies* the diol structure against the product. Picks D.
GPQA_30	Org chem / spec	Color of light absorbed given emission energy 2.34 eV (∼530 nm, green)	Considers BOTH: (a) the textbook Stokes-shift heuristic ("absorbed ≈ emitted, slightly higher energy → blue") AND (b) the *complementary-color* relationship (a green-emitting dye absorbs red, the complementary color). Notes the tension, picks complementary → red. Lands A.	Applies the Stokes-shift heuristic only ("emission at green, absorption at higher energy = blue"). Never considers complementary colors. Picks C.
GPQA_48	Org chem	4-step synthesis ending at a particular alkene; count chemically distinct H atoms on the final structure	Builds the structure step-by-step, then carefully maps the symmetry to count distinct H environments. Lands C (6).	Builds the structure but miscounts the distinct H environments (under-resolves symmetry distinctions). Picks D.
GPQA_98	Org chem	Identify *starting material* given the product of metathesis with methyleneruthenium + 1-propene → 1-(prop-1-en-1-yl)-2-vinylcyclopentane (reverse-inference, again)	Considers all four bicyclic candidates, runs metathesis mechanism on each, eliminates the ones that wouldn't give the observed product. Lands A.	Commits to "1,2-dimethylenecyclopentane gives 1,2-divinyl after metathesis" early; never tests the bicyclic alternatives. Picks D.
GPQA_99	Spec / NMR	Identify compound from FTIR (1700 cm^{-1}, 3000 cm^{-1} broad → carboxylic acid) and 1H NMR (doublet of triplets of quartets, doublet of triplets of triplets)	Decodes the *splitting multiplicities* to count neighbors: dtq = 1+2+3=6 neighbors / specific environments; dtt = 1+2+2=5 neighbors. Maps these to which compound matches. Lands C.	Recognizes carboxylic acid; identifies *one* candidate that "looks reasonable" without working through the splitting-pattern arithmetic. Picks A.
GPQA_103	Org synth	Plan a 7-step synthesis from benzene to 1-(3-bromo-5-nitrophenyl)ethan-1-one; the answer is a specific *order* of reagents	Walks each option's reagent sequence, tracks the directing effects (ortho/para vs meta), confirms which order avoids over-substitution. Lands B.	Identifies the relevant reactions but picks a sequence with wrong ordering; doesn't verify the directing-effect chain. Picks C.
GPQA_111	Org chem	Two Michael additions; predict major products considering *steric hindrance* and *product stability* (literally what the prompt says to consider)	Specifically engages the steric/stability hint: identifies the more-hindered vs less-hindered enolate, the more-stable carbanion. Lands C.	Identifies the addition mechanism but doesn't engage the steric/stability gating. Picks A or B (one product right, one wrong) — fails to combine both considerations.

(continued on next page)

(continued from previous page)

PID	Subject	Question type	What null does right	What COP does wrong
GPQA_132	Spec / NMR	1:1 mixture of two C10H14 aromatics: aromatic H = 2 singlets in 1:1 ratio; methyl H = 3 singlets in 2:1:1 ratio. Identify the two compounds	Goes compound-by-compound (the 4 tetramethylbenzene isomers + 1,4-diethylbenzene), computes expected NMR for each, then checks PAIRS until one matches both constraints. Lands D.	Computes the NMR for ONE pair (option B), finds it "matches perfectly", commits — but misses that the same constraints are also satisfied by D. Picks B (wrong). The model never re-checks the alternative.
GPQA_141	Org chem	Benzyne mechanism on 1-bromobenzene-2-d with $NaNH_2$; count possible organic products. The correct count includes the *isotope-shifted variants* (3 products)	Considers H-abstraction vs D-abstraction routes, tracks the resulting benzyne intermediates, enumerates all three products (aniline, 2-D-aniline, 3-D-aniline). Lands A (3).	Commits to "D is eliminated, only aniline forms" early; or commits to ortho/meta only (2 products). The model doesn't keep both H- and D-abstraction routes open. Picks B/C/D depending on rep.
GPQA_150	Org chem / spec	NMR-characterized C8H9NO + diazotization + heat → identify product. Requires: (a) decode NMR to get tyramine; (b) walk diazonium → phenol; (c) recognize aldol/elimination on the resulting phenol-aldehyde	Combines NMR identification with the multi-step mechanism, lands the conjugated bisphenol enal. Lands A.	Identifies the NMR's tyramine, walks the diazonium step, but doesn't carry through the aldol condensation correctly. Picks an intermediate-stage product.
GPQA_158	Cell bio	Dominant-negative phenotype of a dimerization-domain mutation in a transcription factor	Considers all four phenotypes; recognizes that "protein aggregation + loss-of-function" (B) is THE textbook dominant-negative mechanism for dimerization mutations. Specifically holds onto B as a candidate. Lands B.	Recognizes the dominant-negative concept and commits to D ("degradation of WT") as the textbook answer — confuses the description of effect with the molecular mechanism. The COP output literally reasons "(B) Incorrect. While aggregation can occur, degradation is a more common fate" — applying a generic rule rather than considering that the question is about dimerization-domain mutations specifically, where aggregation IS the most common mechanism.
GPQA_175	Physics	Multipole radiation: spheroidal oscillating charge; find (a) fraction of max power at $\theta = 30°$ and (b) wavelength dependence $f(\lambda)$	Recognizes this is a *multipole* problem (probably quadrupole, given spheroidal), works out $P(\theta) = \sin^2(2\theta)$ for the quadrupole, computes $P(30°)/P_{\max}$, computes λ^{-6} from the quadrupole expression. Lands D.	Applies the *dipole* formula ($P \propto \sin^2\theta$, λ^{-4}), gets fraction = 1/4 and λ^{-4}. Picks A. The model never asks "is this dipole or quadrupole?" — it defaults to dipole, the canonical case.

G.2 The two structural shapes

Reading these 13 cold, one thing comes through that subdomain bucketing would have missed entirely: every one of these problems requires the model to *hold an alternative open* against a strong default interpretation, and then *verify* which one is correct.

Shape 1 — First-plausible-answer trap. The problem cues a specific textbook concept (Stokes shift, Michael addition, dipole radiation, dominant-negative mechanism, name reaction). The model recognizes the

concept instantly. The correct answer requires the model to NOT commit to the cued concept's *standard* completion — it requires considering whether THIS specific case has a twist (complementary color instead of Stokes; quadrupole instead of dipole; aggregation instead of degradation; isotope-shifted products instead of one product). On every one of these, the COP response identifies the canonical concept and writes a confident, well-organized answer that applies the canonical concept's standard interpretation. Null's response, on the same problems, explicitly considers the alternative ("but what if it's quadrupole?", "but the complementary-color rule says red", "but D-abstraction is also possible") before committing.

Shape 2 — Verify-against-each-option. The problem gives four options that look similar (a sequence of reagents, four mechanism products, four NMR patterns). The correct answer requires checking each option against the constraint, not just identifying one that "looks right." On every one of these (GPQA_103, GPQA_132, GPQA_98, GPQA_25, GPQA_19), COP picks the *first* option whose surface features match the problem, without verifying that the OTHER options don't also match (or match better). Null walks all four options, eliminates the bad ones, then commits.

Both shapes share a common cognitive operation: **deliberate consideration of alternatives against a default**. The COP framework — which orients the model toward decisive resolution when an answer is available — short-circuits exactly this operation.

G.3 What the pattern is *not*

The failure mode is not a subdomain failure. The 13 problems span organic chemistry (8), spectroscopy/NMR (2), cell biology (1), physics (1), inorganic (0). The distribution roughly mirrors the benchmark itself.

The failure mode is not a problem-complexity failure either. Null spends 13,325 mean tokens on these 13 problems vs 10,366 across the full benchmark — they're harder than average, but only modestly. COP spends 3,966 mean tokens on these vs 2,581 across the full benchmark. The relative compression (70% vs 75%) is essentially unchanged. The model isn't failing because the problems are too hard; it's failing because COP cuts the *alternatives-considering* phase of reasoning that these specific problems require.

G.4 Data caveats

1. **Sample size.** $N = 13$ high-confidence failures. The shape claim is grounded but not statistically powered. The pattern holds across all 13 on careful read; that is unusual for a noise-dominated effect.
2. **Single-model.** All findings are on Qwen 3.5 27B. The cross-model panel (§4.5) suggests the accuracy cost differs by model; the failure-mode pattern may also differ. Future work: repeat this analysis on Claude Sonnet 4 and Gemma 3 27B failures.
3. **Single-articulation.** The COP condition examined is `DH-revised + C-revised`. The convergence finding (§4.3) shows other articulations track within 1.6pp of accuracy. Whether the same 13 problems fail under other articulations is testable but not tested here.
4. **Researcher judgment.** The per-problem tagging was performed by the listed author. The table above is intentionally granular so a reader can re-check the synthesis. The raw outputs are available in the released JSONL.

G.5 Relaxed set (N=31)

For the broader set of problems where COP loses partial accuracy (null > COP, null $\geq$50% accuracy), the pattern degrades smoothly: the further from the 100%-vs-0% extreme, the less clearly the same shape applies. Among the 18 problems in the relaxed-but-not-high-confidence set, we observe the same first-plausible-answer or verify-against-each-option structure on roughly 12 of them, with the remaining 6 showing more idiosyncratic failure modes. We do not include the relaxed analysis in the paper's main text. The high-confidence set is the load-bearing finding.

Manuscript prepared May 2026. arXiv submission pending. Correspondence: teo@unnma.ai.