

CAVEWOMAN: How Large Language Models Behave Under Linguistic Input and Output Compression

Morayo Danielle Adeyemi
Independent
morayo.danielle@gmail.com

Ryan A. Rossi
Adobe Research
ryrossi@adobe.com

Franck Deroncourt
Adobe Research
franck.deroncourt@adobe.com

Abstract

“Talk short. Drop grammar. Save token.” This caveman style is widely promoted as a way to cut inference cost, but whether it actually saves anything depends on which channel (the user’s prompt or the model’s response) is being compressed. We present CAVEWOMAN, a two-channel evaluation protocol that scores every generation on task accuracy, realized per-item cost, and reference-text agreement against the model’s unconstrained reference. We evaluate eight models on five datasets at five reduction levels, with both channels measured on the same items. Output compression cuts realized cost on most API models (**1.4–2.4**× per model, up to **3**× in the best case) and on all four open-weight models under public-tier pricing. Input compression has the opposite effect, a strict lose-lose: it raises net cost rather than lowering it ($\approx 1.15\times$ on the five-benchmark mean, up to **1.8**× on the worst dataset and **2.7**× under stronger compression), because models compensate with longer responses even as accuracy collapses. Under the same setting, surface text diverges from the unconstrained reference: on the non-reasoning models, roughly half of all generations are correct yet their surface text no longer entails the model’s own unconstrained baseline generation. The divergence survives length-controlled re-scoring, multiple-comparisons correction, and replication under complementary semantic measures. Code and data are available at <https://github.com/danielle34/cavewoman>.

1 Introduction

Inference cost in large language models scales with both input and output token counts, and output tokens are typically priced 4–8× higher than input tokens (Ahia et al., 2023; Nag et al., 2024). Existing compression methods reduce one side or the other, either the prompt (Jiang et al., 2023; Pan et al., 2024; Brussee, 2026; Peltomäki, 2026) or the

response (Xia et al., 2025; Song et al., 2025), but the two are studied separately and both are evaluated almost entirely through task accuracy at a reduced token count. Accuracy at a token count is the wrong instrument for two reasons. It does not separate realized cost from prompt-token reduction (a shorter prompt does not reduce realized cost when the model answers at greater length), and it collapses each generation to a binary outcome that cannot distinguish a compressed answer that agrees with the model’s unconstrained reasoning from one that diverges from it.

We address both gaps with CAVEWOMAN, a two-channel evaluation protocol that scores every generation on three axes: task accuracy, realized per-item cost on the priced channel, and reference-text agreement against the model’s own unconstrained generation. The protocol measures *input compression* (the prompt is filtered before the model sees it) and *output compression* (the model is instructed to answer in a constrained register) at five reduction levels, holding model and item fixed. We evaluate eight models (Qwen2.5-VL-7B, Qwen3.5-9B, DeepSeek-R1-Distill-Qwen-7B, Gemma-4-E4B, GPT-4o, GPT-5.4, Claude Haiku 4.5, Claude Sonnet 4.6) on five benchmarks (GSM8K, BoolQ, ARC-Easy, CommonsenseQA, MMLU-STEM) with complete coverage of both channels at all five levels.

Contributions.

1. We propose a two-channel evaluation protocol that scores compression on realized cost with audited answer-extraction rates and a twelve-metric semantic battery (§3).
2. We measure cost asymmetry between input and output compression on the same items (§4.1).
3. We measure a surface-text divergence between correct answers and the model’s un-

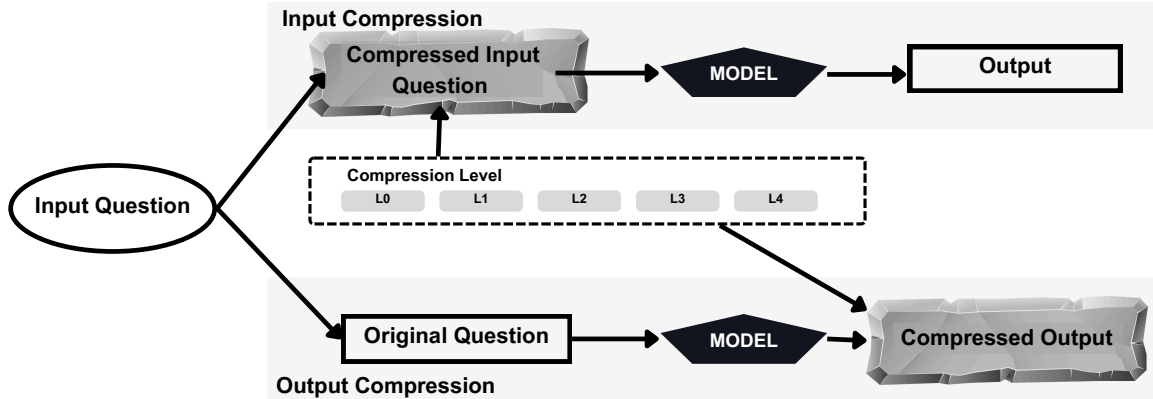


Figure 1: CAVEWOMAN framework. The **input-compression** channel applies a deterministic part-of-speech filter to the user prompt at five reduction levels, leaving the system prompt fixed. The **output-compression** channel leaves the prompt verbatim and replaces the system prompt with a level-specific instruction that requires the same reduction in the response. Every generation is scored on task accuracy, reference-text agreement against the model’s unconstrained reference under bidirectional NLI (plus eleven complementary measures), and per-item input/output token cost.

constrained reference under output compression, replicated across complementary semantic measures (§4.2).

2 Related Work

Input compression. Input compression splits along a hard–soft axis (Li et al., 2025). Hard methods prune tokens by self-information (Li et al., 2023), perplexity (Jiang et al., 2023), question-aware scoring (Jiang et al., 2024), or distilled token classification (Pan et al., 2024), with the same logic at the document level via retrieved-context summarization (Xu et al., 2024). Soft methods encode prompts as gist tokens (Mu et al., 2023), recursive summary vectors (Chevalier et al., 2023), or autoencoder slots (Ge et al., 2024), with rate–distortion bounds in Nagle et al. (2024). None measures whether the response says the same thing it would have without compression.

Output compression. Output-side methods constrain generation via modified decoding (length controls (Kikuchi et al., 2016), budget-signalling positional encodings (Takase and Okazaki, 2019), countdown mechanisms (Song et al., 2025)) or via post-training and prompting (token-skipping chains (Xia et al., 2025), cognitive-inspired routing (Aytes et al., 2025), difficulty-aware prompting (Han et al., 2025), RL-driven demonstration compression (Huang et al., 2024)). The content/function-word split the caveman register exploits is long-standing in linguistics, but this line of work still reports only

task accuracy at a token budget.

Semantic fidelity, verbosity, and cost. Bidirectional NLI entailment separates propositional content from lexical form (Kuhn et al., 2023) and has scored summarization faithfulness (Maynez et al., 2020) and inter-system consistency (Laban et al., 2022); relatedly, chains of thought can be plausible yet causally disconnected from the prediction (Turpin et al., 2023; Lanham et al., 2023). Length itself matters: verbose outputs score lower (Zhang et al., 2025), verbosity carries cost (Borisov et al., 2026), irrelevant padding degrades reasoning (Levy et al., 2024), and chain-of-thought length tracks accuracy independently of trace correctness (Jin et al., 2024; Sun et al., 2025; Wang et al., 2024). The cost-quality frontier is itself a design surface, in selective routing (Chen et al., 2024), quality thresholds (Ding et al., 2024), and preference routers (Ong et al., 2025); valid benchmarking calls for multi-metric measurement (Bowman and Dahl, 2021; Liang et al., 2023; Gehrmann et al., 2021).

Positioning. The closest prior work compresses one side only: LLMLingua (Jiang et al., 2023; Pan et al., 2024; Jiang et al., 2024) on the input; TokenSkip (Xia et al., 2025), Hansel (Song et al., 2025), and Sketch-of-Thought (Aytes et al., 2025) on the output, all reporting task accuracy at compressed token budgets. CAVEWOMAN measures both channels on the same items, reports realized per-item cost rather than token reduction, audits answer-extraction rates before any accuracy claim,

and scores reference-text agreement under complementary semantic criteria. The input-channel divergence reproduces under LLMingua-2 (Appendix D).

3 Methodology

3.1 Experimental Design

Setup. Let \mathcal{M} be a fixed language model and let $x = (w_1, \dots, w_n)$ be a question, a sequence of n tokens from the spaCy tokenizer. Each token w carries a Penn Treebank part-of-speech tag $g(w)$ assigned by spaCy with the `en_core_web_sm` model; g is deterministic given a fixed spaCy version and model.

For a set S of part-of-speech tags, write $(w_i)_{i:g(w_i) \in S}$ for the subsequence of x that retains exactly the tokens whose tag lies in S , with the indices i taken in increasing order. Let $\text{trunc}_k(z)$ denote the prefix of a sequence z of length $\min(|z|, k)$, and let `NN*` and `VB*` denote the Penn Treebank noun and verb tag families.

We study a single reduction parameter, the level $\ell \in \{0, 1, 2, 3, 4\}$, named L0 through L4. The level selects one filter from the family ϕ_0, \dots, ϕ_4 defined by

$$\begin{aligned} \phi_0(x) &= x, \\ \phi_1(x) &= (w_i)_{i:g(w_i) \notin \{\text{DT, IN, CC, RP, TO, MD}\}}, \\ \phi_2(x) &= (w_i)_{i:g(w_i) \in \text{NN}^* \cup \text{VB}^* \cup \{\text{CD}\}}, \\ \phi_3(x) &= (w_i)_{i:g(w_i) \in \text{NN}^* \cup \{\text{CD}\}}, \\ \phi_4(x) &= \text{trunc}_{15}(\phi_3(x)). \end{aligned} \quad (1)$$

The family is nested: for every question x and every $\ell \geq 1$, $\phi_\ell(x)$ is a subsequence of $\phi_{\ell-1}(x)$; a larger ℓ thus applies a stricter reduction. The box below gives the linguistic interpretation of each level.

Two conditions. `CAVEWOMAN` holds \mathcal{M} and x fixed and applies the same reduction ϕ_ℓ at one of two points (Figure 1). The question and the system prompt are formatted with the model’s chat template, held fixed throughout. The L0 baseline is condition-specific: Condition A uses the neutral system prompt below, while Condition B uses the unconstrained step-by-step prompt of Appendix H.2; this difference is by design and later matters for the L0-A/L0-B noise-floor limitation.

Condition A (input compression). The model receives the filtered question $\phi_\ell(x)$ under a neutral system prompt. Condition A tests whether \mathcal{M} needs a full grammatical question.

Condition B (output constraint). The model receives the unmodified question x , and the system prompt instructs it to answer in the reduced form that ϕ_ℓ produces. Prompts are task-neutral: \mathcal{M} infers the answer format from x and a final-line `Answer: <answer>` convention. Condition B tests whether \mathcal{M} needs expressive freedom in its response.

The level ℓ thus indexes both conditions through one family of filters: Condition A reduces what the model reads, and Condition B reduces what the model may write.

3.2 Levels of Linguistic Reduction

We define five linguistic reduction levels, with short names we use throughout the paper: the *unconstrained baseline* (L0), the *telegraphic register* (L1), the *keyword-only register* (L2), the *noun-phrase skeleton* (L3), and the *15-token budget* (L4). Each level removes further word classes from the previous level, forming a monotone hierarchy in which a change in accuracy between adjacent levels is attributable to the words removed at that step. The POS filter is a deterministic, transparent reduction chosen for measurement clarity; learned compressors with task-aware token scoring (Appendix D) make different trade-offs.

Reduction Levels

L0: Unconstrained baseline. ϕ_0 is the identity and keeps every tag. Condition A presents x verbatim, and Condition B asks for step-by-step reasoning in full sentences.

L1: Telegraphic. ϕ_1 removes the closed-class tags `DT`, `IN`, `CC`, `RP`, `TO`, and `MD`: determiners, prepositions, conjunctions, particles, infinitival *to*, and modals.

L2: Keyword only. ϕ_2 keeps only nouns, verbs, and cardinal numbers.

L3: Noun-phrase skeleton. ϕ_3 keeps only nouns and cardinal numbers.

L4: 15-token budget. ϕ_4 applies ϕ_3 and then truncates the result to its first 15 tokens.

The same five levels apply to both conditions but enter the pipeline at different points. In Condition A (input compression), ϕ_ℓ rewrites the user message via the deterministic POS-tag filter. In Condition B (output constraint), a level-specific

system prompt instructs the model to produce its response in the matching register; the user message is left intact. The per-level decoder budget is $\text{max_new_tokens} \in \{400, 300, 200, 150, 20\}$ across L0–L4, identical for both conditions. Both the verbatim per-level filter rules (Condition A) and the verbatim per-level system prompts (Condition B) are listed in Appendix H. L4 is excluded from reference-text-agreement scoring since its Condition B system prompt asks the model to emit only the answer; the 15-token budget is conveyed as a prompt instruction rather than a hard decoder cap, and its soft-enforcement details are in Appendix G. A worked example of one question filtered through all five Condition A levels is in Appendix H (Figure 8).

3.3 Evaluation Metrics

We score every generation on three axes: task accuracy (the regex-extracted answer matches the item’s ground-truth answer, with a 0.01 tolerance on numeric answers), per-item realized token cost on the priced channel, and reference-text agreement against the model’s unconstrained generation on the same item and channel. Reference-text agreement is operationalized by bidirectional NLI entailment, with a DeBERTa-based NLI judge (He et al., 2021) as the conservative headline criterion, and replicated under eleven complementary semantic criteria (Appendix C).

3.4 Dissociation Table

For each level L_x we build a 2×2 table that crosses task correctness with *reference-text agreement against L0* (Table 1). C_2 and C_3 are distinct outcomes that accuracy-only evaluation cannot separate; outcome shares are robust to metric choice (Appendix C).

	Entails L0	Does not entail L0
Correct	C_1	C_2
Incorrect	C_3	C_4

Table 1: 2×2 dissociation table, applied at L1–L3 (L4 excluded from semantic evaluation; see §3.2). “Entails L0” means bidirectional NLI entailment against the *same-channel* L0 reference. C_2 is correct answers paired with surface-text divergence from the same-channel L0 reference; C_3 is reference-text agreement despite an incorrect answer. Accuracy-only evaluation cannot separate these outcomes. Operationalized by bidirectional NLI in the main text and replicated under the alternative criteria of Appendix C.

3.5 Datasets

We use five datasets spanning four task types and three answer formats (Table 2): GSM8K (Cobbe et al., 2021), BoolQ (Clark et al., 2019), ARC-Easy (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), and the STEM split of MMLU (Hendrycks et al., 2021). BoolQ and CommonsenseQA use validation splits; GSM8K, ARC-Easy, and MMLU-STEM use test splits; MMLU is restricted to 20 STEM subjects.

Dataset	n	Answer type
GSM8K	1,319	numeric
BoolQ	3,270	boolean
ARC-Easy	2,376	MC (A–D)
CommonsenseQA	1,221	MC (A–E)
MMLU-STEM	3,279	MC (A–D)

Table 2: Datasets used in CAVEWOMAN, spanning math word problems (GSM8K), passage yes/no (BoolQ), science multiple-choice (ARC-Easy), commonsense multiple-choice (CommonsenseQA), and STEM multiple-choice (MMLU-STEM). A single model is evaluated on 11,465 items at five reduction levels under both conditions.

3.6 Models

We evaluate Qwen2.5-VL-7B (Bai et al., 2025), Qwen3.5-9B (Yang et al., 2025), DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025), Gemma-4-E4B (Google, 2026), GPT-4o (Hurst et al., 2024), GPT-5.4 (OpenAI, 2026), Claude Haiku 4.5 (Anthropic, 2025), and Claude Sonnet 4.6 (Anthropic, 2026). All eight models are evaluated on every benchmark under both conditions. Qwen2.5-VL-7B is included on these text-only benchmarks because the evaluation uses its text backbone in ordinary chat mode rather than any vision input path. Two reasoning-protocol details affect interpretation: DeepSeek-R1 emits hidden `<think>` traces that count against the same token budget as its visible output, and Qwen3.5-9B has an optional thinking mode that we leave off (the model default). We nonetheless group Qwen3.5-9B with the reasoning models because, like the distilled reasoner, its unconstrained generations are already short and terse, the property that governs how much surface text can diverge under output compression. Mechanics and the exclusion of a third reasoning model (Kimi-K2.6 (Kimi Team, 2026)) are in Appendix E; full inference configuration is in Appendix A.3.

4 Results

4.1 Finding 1: Cost asymmetry between channels

Finding 1. Output compression cuts realized cost on most API models ($1.4\text{--}2.4\times$ per model, up to $3\times$ on the best cell; cheaper on 17 of 20 ‘(model, dataset)’ cells) and all four open-weight models. Input compression instead raises net cost through compensatory output expansion: up to $1.15\times$ on the five-benchmark mean and $1.8\times$ on the worst individual dataset, growing to $2.7\times$ at deeper reductions as accuracy collapses.

Per-item cost for an API-served model is $C = n_{\text{in}} p_{\text{in}} + n_{\text{out}} p_{\text{out}}$, where n_{in} and n_{out} are the input and output token counts and p_{in} , p_{out} the corresponding per-token prices (May 2026). Figure 3 reports the relative change in realized cost against the same-channel unconstrained baseline. Output compression reduces realized per-item cost on GPT-4o, Claude Haiku 4.5, and Claude Sonnet 4.6 by the per-model margins above, cheaper on every one of their fifteen benchmarks; the exception is GPT-5.4, whose billed output is dominated by hidden reasoning tokens (Appendix E), so it is cheaper on only two of its five benchmarks (17 of 20 API cells in all). Input compression at the same reduction levels raises net cost on the remaining models (GPT-4o aside) before accuracy collapses at the strictest level (Figure 2). This is a strict lose-lose: the telegraphic level already raises net cost (up to $1.8\times$ on individual datasets) while degrading accuracy, and at deeper reduction levels the worst-case penalty grows to $\sim 2.7\times$ as accuracy collapses to single digits; the input channel raises cost and lowers accuracy at the same time.

The mechanism is compensatory output expansion. Stripping function words from the prompt saves a small number of input tokens, but the model answers at greater length and output tokens cost several times more than input tokens; the net change is positive. Every API model saves input tokens but spends more on output, and the priced ratio between the two leaves net cost higher on every API model but GPT-4o, where the two effects roughly cancel (Table 3). Output compression instead saves on the output side, which dominates cost: the most favorable case (GPT-4o on ARC-Easy) is roughly three times cheaper at the same

Model	Δ in tok	Δ out tok	Net cost
GPT-4o	-15.3	+0.1	-2.6%
GPT-5.4	-15.3	+18.0	+15.4%
Claude Haiku 4.5	-15.7	+9.9	+3.1%
Claude Sonnet 4.6	-15.7	+12.9	+5.1%

Table 3: Per-model token economics under input compression at the telegraphic level, averaged across the five benchmarks. Input tokens fall on every model; output tokens rise on every model; net cost is positive on all but one configuration.

accuracy, and on GSM8K the same model trades a small accuracy gain against a halving of cost. The saving requires that the priced output actually shrink under the constraint, which holds for every model whose billed output matches its visible response.

Apparent accuracy gains can be a parser artifact. Answer-extraction rates can inflate apparent accuracy gains under compression. MMLU-STEM provides the clearest example, several API models have lower answer-extraction rates under the unconstrained setting than under compressed output, so apparent improvements partly reflect easier parsing rather than better reasoning. To avoid attributing parser recovery to the model itself, we only report “compressed exceeds unconstrained” gains when the unconstrained extraction rate is at least 0.95, and flag the remaining cases as extractor-suppressed (Appendix G).

Scope of the cost win. Output-channel cost savings are largest on benchmarks whose ground-truth answer is already short (BoolQ yes/no, MCQ-letter): the L1 instruction collapses the response to roughly the answer plus minimal reasoning, and savings are partly mechanical. The per-model savings (mean $1.4\text{--}2.4\times$ on the three models that save) should therefore be read against this format-collapse component; on GSM8K, where the answer requires multi-step arithmetic, savings are smaller but accuracy is preserved on the cells with ≥ 0.95 L0 parse rate. Apparent single-cell gains on MMLU-STEM (e.g. Gemma-4-E4B, +35.3 pp) are confounded by answer-extraction recovery (the L0-B parse rate rises from 0.49 to 0.88 at L1) and are reported as extractor-suppressed rather than reasoning gains (Appendix G).

Under public-tier pricing, all four open-weight models also save cost at L1 under output compression, in the same direction as the API panel (Appendix A.6).

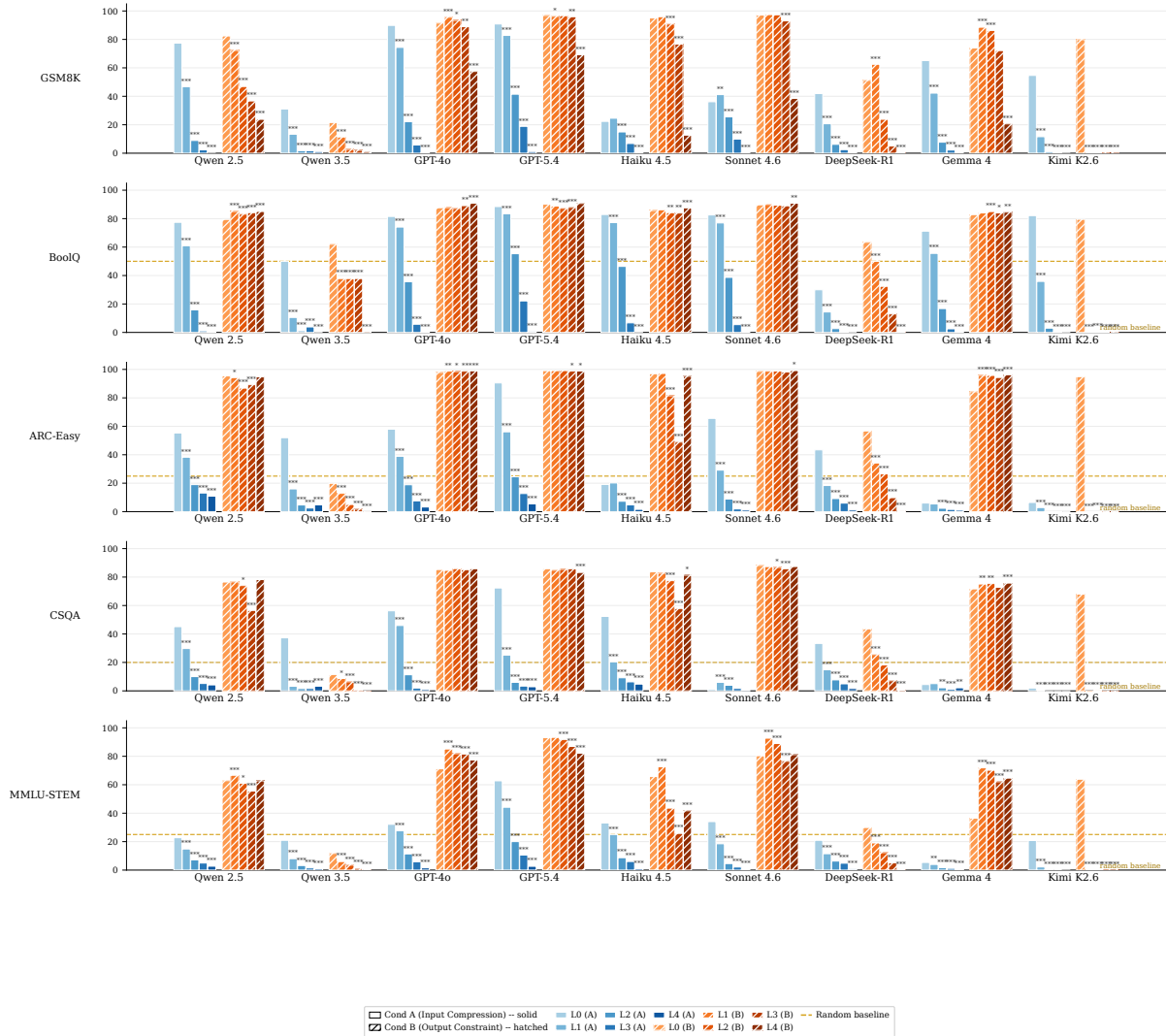


Figure 2: Answer accuracy across the five reduction levels for all models and benchmarks. Solid bars denote input compression; hatched bars denote output compression. Significance markers indicate Wilcoxon signed-rank tests against the within-model unconstrained baseline ($*p < .05$, $**p < .01$, $***p < .001$). The dashed gold line marks the random-guessing baseline for each benchmark. Kimi-K2.6 is shown for completeness only and is excluded from all aggregates (Appendix E).

4.1.1 Channel-specific degradation

The two channels also degrade differently. Under input compression, accuracy and reference-text agreement fall together as the level increases; under output compression, classification accuracy holds through deep levels while reference-text agreement falls sharply at the first level. The input channel spends accuracy; the output channel spends agreement with the unconstrained reference. The output-channel pattern motivates Finding 2.

4.2 Finding 2: Accuracy decouples from same-channel reference text under output compression

Finding 2. Across the six non-reasoning models, 51.9% of all L1 output-compression generations are correct yet have surface text that no longer matches what the model would have written without the constraint on the six-non-reasoning panel; under length-matched re-scoring on the same panel, the rate rises to 80.4%.

The headline rate is the share of all generations at L1 output compression that are correct yet

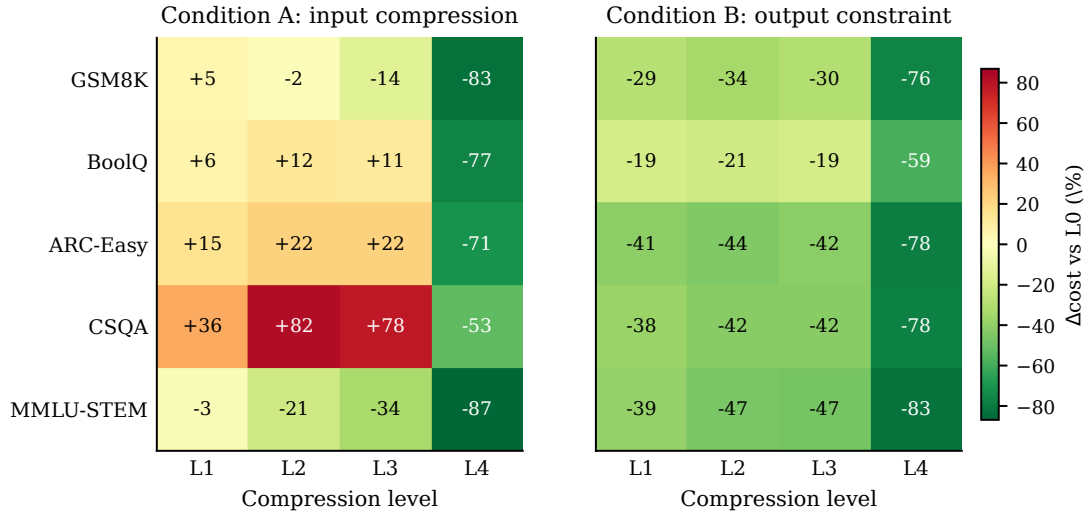


Figure 3: Relative change in estimated per-item inference cost against the unconstrained baseline, averaged across the four API models. Left panel: input compression. Right panel: output compression. Rows are benchmarks; columns are the four non-zero reduction levels. Red denotes a cost increase, green a cost reduction. The two channels move in opposite directions at the same reduction level. Worst-case input-channel penalties reach $1.8\times$ at L1 and $\sim 2.7\times$ at deeper reductions (Finding 1).

no longer entail the same-channel unconstrained reference under a bidirectional NLI judge (the C_2 cell of Table 1). Pooled across the six non-reasoning models in our panel, the rate is 51.9% on the six-non-reasoning panel. The two reasoning models in the panel (DeepSeek-R1-Distill and Qwen3.5-9B) show a smaller divergence since their unconstrained generations are already short (per-model values in Appendix C.3). On every non-reasoning model, the dominant off-diagonal outcome is correct-but-divergent rather than incorrect-but-faithful.

Accuracy holds while the reasoning trace drifts. Figure 4 resolves L1 output compression into its 2×2 outcome cells per model. The amber C_2 segment is the dissociation: correct answers whose surface text no longer entails the model’s same-channel L0 reference. On every non-reasoning model the C_2 band is the dominant off-diagonal cell; DeepSeek-R1 inverts the pattern (Appendix E).

Length is not the explanation. Length-matched re-scoring (truncating L0 to the L1-B wordpiece-token length) *increases* the divergence on every non-reasoning model (Table 16, Appendix C.3); the headline is the conservative reading.

Robustness across judges, metrics, and statistics. The bidirectional NLI judge has a calibrated

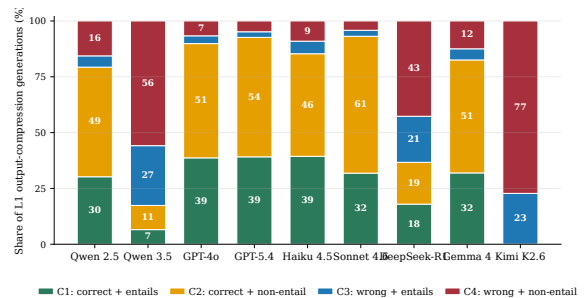


Figure 4: Per-model 2×2 dissociation at L1 output compression, summed across the five benchmarks. Each stacked bar partitions L1 output-compression generations into the four outcome cells: C_1 (correct + entails L0), C_2 (correct + does not entail L0), C_3 (wrong + entails L0), and C_4 (wrong + does not entail L0). Accuracy is C_1+C_2 and bidirectional NLI rate is C_1+C_3 ; the amber C_2 share is the dissociation. DeepSeek-R1 inverts the pattern. Kimi-K2.6 is shown for completeness only and is excluded from all aggregates (Appendix E).

false-negative rate of 2.9% at L1 on synthetic positive pairs (Appendix C.1); all twelve semantic measures report a substantial divergence, with our headline sitting near the conservative end of the family (Table 15). Wilcoxon signed-rank tests against the within-model, within-channel unconstrained baseline survive Benjamini–Hochberg correction at $\alpha = 0.05$ on every significant cell (Appendix F).

Generalization and task-type dependence. The divergence also holds under the learned LLMingua-2 compressor (Pan et al., 2024): $C_2 >$

0 on every cell at $\tau=0.5$, with rate-driven recovery at $\tau=0.8$ (Appendix D). The divergence is smaller on arithmetic benchmarks than on classification (Table 13). Register traces across L1–L4 are in Appendix I.

4.3 Robustness to compression varies across models

Finding 3. Relative robustness (telegraphic output- vs. input-constraint accuracy) spans a $1.5\times$ to $3.7\times$ range across models and is not predicted by parameter count or unconstrained accuracy.

Let robustness be the ratio of accuracy under the telegraphic output constraint to accuracy under the telegraphic input constraint, at the same item and the same reduction level. The ratio is reported per model in Table 4. The most output-robust model overall is the open-weight Gemma-4-E4B, and the most output-robust API model (Claude Sonnet 4.6) retains nearly three times as much accuracy under the output constraint as under the input constraint; the highest-accuracy API model on the unconstrained baseline (GPT-5.4) is least robust among the four API models (second-least across all eight). Gemma-4-E4B tops the full ranking, and Qwen2.5-VL-7B outranks two of the four API models. DeepSeek-R1’s value should be read with the reasoning-token caveat of Appendix E, since its hidden chain-of-thought tokens are charged against the same visible-output budget; with two API vendors at two models each, we draw no vendor-level conclusion. The practical implication is that a deployment should rank candidates at the constraint level under which it will be deployed, not at the unconstrained baseline.

Model	Output / input acc. ratio
Gemma-4-E4B	3.70
Claude Sonnet 4.6	2.71
Claude Haiku 4.5	2.60
DeepSeek-R1	2.40
Qwen2.5-VL-7B	2.07
GPT-4o	1.73
GPT-5.4	1.59
Qwen3.5-9B	1.50

Table 4: Output-vs-input accuracy ratio at the telegraphic level, per model. Higher values indicate greater robustness to output compression. Parameter count does not predict the ordering.

5 Discussion

Output is the channel to compress on when the model answers at length (Finding 1). It produces real cost savings on every API model whose billed output matches its visible response, while input compression does not.

Robustness to output compression does not follow from unconstrained accuracy. The highest-accuracy API model in our panel is least robust among the four API models (second-least across all eight), and a 7B open-weight model outranks two of the four API models under the output constraint.

The accuracy/reference-text divergence is a surface-text observation, not a propositional claim. Findings 1 and 2 answer different questions and can both hold: correctness is graded against the ground-truth answer, while reference-text agreement is graded against the model’s own L0 generation. For deployments that consume only the final answer, the divergence does not affect outcomes; for deployments that consume the generation as text (transcripts, audit trails, reasoning displays), it does.

These findings suggest that single-axis evaluation of compression is underdetermined. Realised cost on the priced channel is not reducible to prompt-token reduction; observed accuracy is conditional on extraction-rate reliability; and reference-text agreement is conditional on the choice of semantic axis. The three quantities dissociate in our panel; any composite metric that aggregates them can therefore mis-rank methods on the dimension that ultimately determines deployment cost.

6 Conclusion

Compressing language-model inference is a two-channel problem, and accuracy alone cannot tell the channels apart. Output compression cuts realized cost on most API models ($1.4\text{--}2.4\times$ per model, up to $3\times$ on the best cell, at the first reduction level) and all four open-weight models under public-tier pricing; input compression instead raises net cost through compensatory output expansion (up to $\sim 15\%$ on the five-benchmark mean and $1.8\times$ on individual datasets, growing to $2.7\times$ at deeper reductions as accuracy collapses). On the same cost-saving settings, 51.9% of generations on the six-non-reasoning panel are correct yet their surface text no longer matches the model’s unconstrained reference, a divergence that strengthens

under length-controlled re-scoring and replicates across complementary semantic measures. Robustness to output compression varies widely across models and is not predicted by parameter count or unconstrained accuracy; candidates should therefore be ranked at the constraint level they will be deployed at, not at the unconstrained baseline.

6.1 Limitations

Our bidirectional NLI judge measures surface-text divergence rather than propositional drift; we bound the length-and-register confound through length-controlled re-scoring (Appendix C.3) and replicate under eleven complementary measures plus the headline judge (Appendix C), but the rate is not a propositional-content claim. Part of the Condition B divergence may also reflect the system-prompt register change between conditions; the L0-A/L0-B noise floor that would isolate this is not separately measured (Appendix C.3). All five benchmarks have short, structured answers (numeric, boolean, MCQ letter); we make no claim about content preservation in long-form generation tasks such as summarization or open-ended QA. The eight-model panel uses greedy decoding only, with two API vendors at two models each; we therefore draw no vendor- or family-level conclusion; sampled decoding and a hard-decoder L4 are out of scope. Full per-confound discussion is in Appendix F.

Acknowledgments

We thank MIT Engaging for providing the GPU compute used for the local-model runs.

References

- Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A. Smith, and Yulia Tsvetkov. 2023. [Do all languages cost the same? tokenization in the era of commercial language models](#). In *The 2023 Conference on Empirical Methods in Natural Language Processing*.
- Anthropic. 2025. [System card: Claude haiku 4.5](#). Technical report, Anthropic. System card documenting improvements and model safety testing for Claude Haiku 4.5.
- Anthropic. 2026. [System card: Claude sonnet 4.6](#). Technical report, Anthropic. Technical system card documenting model safeguards, model characteristics, and deployment of Claude Sonnet 4.6.
- Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24307–24331.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuezhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.
- Vadim Borisov, Michael Gröger, Mina Mikhael, and Richard H Schreiber. 2026. Do chatbot llms talk too much? the yapbench benchmark. *arXiv preprint arXiv:2601.00624*.
- Samuel Bowman and George Dahl. 2021. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855.
- Julius Brusse. 2026. Caveman. <https://github.com/juliusbrussee/caveman>. GitHub repository.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2024. [FrugalGPT: How to use large language models while reducing cost and improving performance](#). *Transactions on Machine Learning Research*. Featured Certification.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2924–2936.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhadrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. 2024. Hybrid llm:

- Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, volume 2024, pages 41348–41366.
- Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. [In-context autoencoder for context compression in a large language model](#). In *The Twelfth International Conference on Learning Representations*.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh Dhole, and 1 others. 2021. The gem benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120.
- Google. 2026. Gemma 4: Our most capable open models to date. <https://blog.google/innovation-and-ai/technology/developers-tools/gemma-4/>. Google Blog.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [{DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}](#). In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Xijie Huang, Li Lyna Zhang, Kwang-Ting Cheng, Fan Yang, and Mao Yang. 2024. Fewer is more: Boosting math reasoning with reinforced context pruning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13674–13695.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. [LLMLingua: Compressing prompts for accelerated inference of large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.
- Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenye Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.
- Kimi Team. 2026. [Kimi k2.6 technical report](#).
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. [Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation](#). In *The Eleventh International Conference on Learning Representations*.
- Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.
- Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353.
- Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6342–6353.
- Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human*

- Language Technologies (Volume 1: Long Papers)*, pages 7182–7195.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. [Holistic evaluation of language models](#). *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1906–1919.
- Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352.
- Arijit Nag, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2024. [Cost-performance optimization for processing low-resource language tasks using commercial LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15681–15701, Miami, Florida, USA. Association for Computational Linguistics.
- Alliot Nagle, Adway Girish, Marco Bondaschi, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. 2024. Fundamental limits of prompt compression: A rate-distortion framework for black-box language models. *Advances in Neural Information Processing Systems*, 37:94934–94970.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. [RouteLLM: Learning to route LLMs from preference data](#). In *The Thirteenth International Conference on Learning Representations*.
- OpenAI. 2026. Introducing GPT-5.4: Designed for professional work. <https://openai.com/index/introducing-gpt-5-4/>. OpenAI Blog.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, and 1 others. 2024. LlmLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.
- William Peltomäki. 2026. Caveman compression. <https://github.com/wilpel/caveman-compression>. GitHub repository.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Seoha Song, Junhyun Lee, and Hyeonmok Ko. 2025. Hansel: Output length controlling framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 23, pages 25146–25154.
- Yi Sun, Han Wang, Jiaqiang Li, Jiacheng Liu, Xiangyu Li, Hao Wen, Yizhen Yuan, Huiwen Zheng, Yan Liang, Yuanchun Li, and 1 others. 2025. An empirical study of llm reasoning ability under strict output length constraint. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7663–7682.
- Sho Takase and Naoaki Okazaki. 2019. [Positional encoding to control output sequence length](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.
- Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. 2024. Reasoning in token economies: budget-aware evaluation of llm reasoning strategies. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19916–19939.
- Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3351–3363.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. Re-comp: Improving retrieval-augmented llms with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pages 43478–43502.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others.

2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yusen Zhang, Sarkar Snigdha Sarathi Das, and Rui Zhang. 2025. Demystify verbosity compensation behavior of large language models. In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertainNLP 2025)*, pages 160–178.

A Implementation Details

Pricing, dataset statistics and licenses, inference configuration, token accounting, cost estimates, and the L1 decoder-truncation check. Verbatim system prompts and POS-tag rules are in Appendix H; released artifacts in Appendix J.

A.1 Pricing Assumptions and Model Snapshots

May 2026 per-token API prices for the four closed models (Table 5). Per-item API cost is $C = n_{in} \cdot p_{in} + n_{out} \cdot p_{out}$, with tokens counted by each model’s own tokenizer. Open-weight models run on local GPUs and are excluded from the dollar-cost analysis; token accounting is in Appendix B.

A.2 Dataset Statistics and Licenses

Per-dataset sizes, splits, answer formats, mean L0 token lengths, and licenses are in Table 6.

A.3 Inference Configuration

All eight models use greedy decoding under identical settings for both conditions; per-level decoder budgets are the only level-dependent parameter (Table 7).

A.4 Token Accounting

Mean tokens per level for Claude Haiku 4.5, Condition A, CommonsenseQA (Table 8). Input tokens fall at L1–L2 but output expansion produces a net increase at L1 and marginal savings at L2; reductions appear at L3–L4, where entailment has already declined.

A.5 Cost Estimates

Cost model applied to Claude Haiku 4.5, Condition A, CommonsenseQA (Table 9). Cost rises at L1 from output expansion; reductions appear only at higher levels where entailment has fallen.

Per-model, per-benchmark breakdowns are in Appendix B; the input/output token split reproduces from the per-token rates of Table 5.

A.6 Open-Weight Cost Projection

The four open-weight models (Qwen2.5-VL-7B, Qwen3.5-9B, DeepSeek-R1-Distill-Qwen-7B, Gemma-4-E4B) have no metered API price, so we project their measured input/output token counts onto a six-tier panel of May 2026 public pricing (DSv4 Flash, DSv4 Pro, Haiku 4.5, GPT-4o, GPT-5.4, Sonnet 4.6) and average across tiers. At L1 Cond B the mean projected savings are $2.5\times$ on Qwen2.5-VL-7B (the largest open-weight saving in the panel), $1.18\times$ on Qwen3.5-9B, $1.17\times$ on DeepSeek-R1-Distill, and $2.09\times$ on Gemma-4-E4B, all in the same direction as the API panel; the two reasoning-distilled models save less because their unconstrained generations are already short. Full per-tier, per-cell numbers are released in our repository.

A.7 Decoder-Truncation Check at L1 Condition B

Fraction of items whose output hit the L1 `max_new_tokens=300` ceiling: 0.9% (GPT-4o), 7.3% (Haiku 4.5), 0.9% (Qwen2.5-VL-7B). The L1-B cost savings of Finding 1 reflect natural stopping, not truncation.

Model	Input (\$/M tokens)	Output (\$/M tokens)	Snapshot / endpoint
GPT-4o	2.50	10.00	gpt-4o-2024-11-20
GPT-5.4	1.25	10.00	gpt-5-2025-08-07
Claude Haiku 4.5	1.00	5.00	claude-haiku-4-5-20251001
Claude Sonnet 4.6	3.00	15.00	claude-sonnet-4-6-20250929
Qwen 2.5-VL-7B	local GPU		Qwen/Qwen2.5-VL-7B-Instruct
Qwen 3.5-9B	local GPU		Qwen/Qwen3.5-9B
DeepSeek-R1-Distill 7B	local GPU		deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
Gemma-4-E4B	local GPU		google/gemma-4-e4b

Table 5: Per-token API pricing (USD/M tokens, May 2026) and model snapshots. The four open-weight models run on 2× NVIDIA L40S 48GB at bfloat16; the full open-weight evaluation across all five benchmarks and both channels took approximately 425 GPU-hours.

Dataset	<i>n</i>	Split	Format	Mean toks	License
GSM8K	1,319	test	numeric	64	MIT
BoolQ	3,270	val	yes/no	117	CC-BY-SA-3.0
ARC-Easy	2,376	test	MC (A–D)	49	CC-BY-SA-4.0
CommonsenseQA	1,221	val	MC (A–E)	33	MIT
MMLU-STEM	3,279	test	MC (A–D)	85	MIT

Table 6: Dataset statistics. Mean tokens at L0 under the GPT-4o tokenizer. All datasets used under their published licenses.

Setting	Value
Decoding	greedy (temperature = 0, do_sample=False)
max_new_tokens (L0–L4)	{400, 300, 200, 150, 20}
spaCy	3.8.14 with en_core_web_sm 3.8.0
Open-weight hardware	2× NVIDIA L40S 48GB, bfloat16 (~425 GPU-hours total)
NLI judge	cross-encoder/nli-deberta-v3-base
Embedding model (cosine)	sentence-transformers/all-MiniLM-L6-v2 (Reimers and Gurevych, 2019)
Random seed	42 (where applicable)

Table 7: Inference configuration. Identical across Conditions A and B.

Level	Mean in	Mean out	Total	Out/In	Total Δ
L0	62.8	170.3	233.1	2.71	baseline
L1	53.6	192.4	246.0	3.59	+5.5%
L2	40.7	180.9	221.6	4.44	-4.9%
L3	36.5	146.7	183.2	4.02	-21.4%
L4	35.9	20.0	55.9	0.56	-76.0%

Table 8: Token accounting for Claude Haiku 4.5, Condition A, CommonsenseQA ($n = 1,221$). Total tokens rise at L1 from output expansion.

Lv	In cost	Out cost	Total cost	Cost Δ	NLI%
		(\$/M items)			
L0	62.8	851.5	914.3	baseline	100.0
L1	53.6	962.0	1015.6	+11.1%	50.8
L2	40.7	904.5	945.2	+3.4%	36.4
L3	36.5	733.5	770.0	-15.8%	21.9
L4	35.9	100.0	135.9	-85.1%	—

Table 9: Estimated inference cost per million items for Claude Haiku 4.5, Condition A, CommonsenseQA. Lower cost does not imply higher preservation.

B Per-Level Results Tables

Per-level accuracy and bidirectional NLI for all eight models (Tables 10–11). Colored deltas show the change against each model’s L0 baseline (red \downarrow degradation, green \uparrow improvement). Green Acc deltas with red NLI deltas on classification benchmarks under Condition B quantify Finding 2; DeepSeek-R1 shows the inverse pattern (§4.2). Figure 5 visualizes the accuracy data as per-model, per-dataset curves.

B.1 Threshold and Task-Type Comparisons

Smallest level L_c at which accuracy (L_c^{acc}) and NLI (L_c^{sem}) cross the degradation criterion (Table 12): we define degradation as accuracy falling at least 5 pp below the L0 baseline (L_c^{acc} , scanned over L1–L4) and bidirectional NLI falling at least 15 pp below the L0 anchor (L_c^{sem} , scanned over L1–L3 since L4 carries no NLI score); “—” marks a cell where the threshold is never crossed in that range (non-evaluable for the ordering). Of 80 ‘(model, dataset, condition)’ cells, 60 are evaluable (both thresholds crossed in range), and on the 8-model panel under strict bidirectional NLI, $L_c^{\text{sem}} \leq L_c^{\text{acc}}$ on all 60. Forward-only NLI violates the ordering on 2 of 60; cosine on 36 of 60, confirming cosine is non-monotone. The rows in Table 12 are a representative three-model excerpt; the 60-of-60 figure is computed over the full eight-model panel (released data). Table 13 aggregates L1 non-preservation by benchmark; Figure 6 breaks the 2×2 dissociation down per-benchmark.

Model	Lv	GSM8K		BoolQ		ARC-Easy		CSQA		MMLU-STEM	
		Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑
Qwen2.5-VL-7B	L0	77.4	100.0	77.4	100.0	55.3	100.0	45.1	100.0	22.7	100.0
	L1	46.8(30.6↓)	71.2(28.8↓)	60.9(16.5↓)	22.8(77.2↓)	38.3(17.0↓)	23.1(76.9↓)	29.7(15.4↓)	25.2(74.8↓)	14.8(7.9↓)	30.8(69.2↓)
	L2	8.9(68.5↓)	44.9(55.1↓)	15.9(61.5↓)	10.8(89.2↓)	19.0(36.3↓)	10.7(89.3↓)	10.0(35.1↓)	16.1(83.9↓)	7.2(15.5↓)	16.7(83.3↓)
	L3	2.4(75.0↓)	27.7(72.3↓)	1.0(76.4↓)	7.2(92.8↓)	13.1(42.2↓)	6.6(93.4↓)	5.2(39.9↓)	7.3(92.7↓)	4.9(17.8↓)	10.2(88.8↓)
	L4	0.0(77.4↓)	—	0.2(77.2↓)	—	10.9(44.4↓)	—	4.2(40.9↓)	—	2.7(20.0↓)	—
Qwen3.5-9B	L0	31.1	100.0	50.0	100.0	52.0	100.0	37.3	100.0	20.9	100.0
	L1	13.3(17.8↓)	48.6(51.4↓)	10.5(39.5↓)	50.9(49.1↓)	16.0(36.0↓)	25.2(74.8↓)	3.2(34.1↓)	23.3(76.7↓)	8.0(12.9↓)	31.0(69.0↓)
	L2	1.7(29.4↓)	37.5(62.5↓)	1.1(48.9↓)	35.4(64.6↓)	4.7(47.3↓)	15.4(84.6↓)	1.6(35.7↓)	20.9(79.1↓)	3.2(17.7↓)	14.6(85.4↓)
	L3	1.7(29.4↓)	30.1(69.9↓)	3.9(46.1↓)	31.4(68.6↓)	2.5(49.5↓)	13.2(86.8↓)	1.6(35.7↓)	20.1(79.9↓)	1.5(19.4↓)	11.4(88.6↓)
	L4	1.1(30.0↓)	—	0.0(50.0↓)	—	4.8(47.2↓)	—	3.3(34.0↓)	—	1.0(19.9↓)	—
GPT-4o	L0	89.8	100.0	81.5	100.0	58.0	100.0	56.3	100.0	32.1	100.0
	L1	74.4(15.4↓)	81.0(19.0↓)	74.1(7.4↓)	35.6(64.4↓)	38.9(19.1↓)	30.1(69.9↓)	45.9(10.4↓)	30.9(69.1↓)	27.7(4.4↓)	44.9(55.1↓)
	L2	22.1(67.7↓)	55.0(45.0↓)	35.7(45.8↓)	16.4(83.6↓)	18.9(39.1↓)	15.4(84.6↓)	11.3(45.0↓)	16.1(83.9↓)	11.4(20.7↓)	21.8(78.2↓)
	L3	5.8(84.0↓)	39.3(60.7↓)	5.7(75.8↓)	7.1(92.9↓)	7.6(50.4↓)	7.7(92.3↓)	2.0(54.3↓)	5.1(94.9↓)	5.8(26.3↓)	12.5(87.5↓)
	L4	0.4(89.4↓)	—	0.0(81.5↓)	—	3.2(54.8↓)	—	1.0(55.3↓)	—	1.6(30.5↓)	—
GPT-5.4	L0	90.8	100.0	88.5	100.0	90.5	100.0	72.3	100.0	62.7	100.0
	L1	82.8(8.0↓)	68.3(31.7↓)	83.5(5.0↓)	39.5(60.5↓)	56.1(34.4↓)	34.6(65.4↓)	25.1(47.2↓)	28.9(71.1↓)	44.2(18.6↓)	50.4(49.6↓)
	L2	41.5(49.4↓)	40.2(59.8↓)	55.3(33.1↓)	23.1(76.9↓)	24.5(66.0↓)	21.0(79.0↓)	6.0(66.3↓)	16.8(83.2↓)	20.0(42.7↓)	29.6(70.4↓)
	L3	18.9(71.9↓)	24.6(75.4↓)	22.1(66.3↓)	18.1(81.9↓)	12.7(77.8↓)	15.2(84.8↓)	3.4(69.0↓)	9.8(90.2↓)	10.6(52.2↓)	23.3(76.7↓)
	L4	1.1(89.7↓)	—	0.7(87.8↓)	—	5.5(85.0↓)	—	2.9(69.5↓)	—	2.7(60.1↓)	—
Haiku 4.5	L0	22.4	100.0	82.9	100.0	19.1	100.0	52.3	100.0	33.1	100.0
	L1	24.6(2.2↑)	62.8(37.2↓)	77.2(5.7↓)	37.2(62.8↓)	20.2(1.1↑)	46.3(53.7↓)	20.4(31.9↓)	50.8(49.2↓)	25.0(8.1↓)	44.0(56.0↓)
	L2	14.9(7.5↓)	29.7(70.3↓)	46.4(36.5↓)	20.8(79.2↓)	7.3(11.8↓)	29.4(70.6↓)	9.2(43.1↓)	36.4(63.6↓)	8.7(24.4↓)	25.7(74.3↓)
	L3	6.8(15.6↓)	19.0(81.0↓)	6.7(76.2↓)	18.9(81.1↓)	4.8(14.3↓)	19.7(80.3↓)	6.3(46.0↓)	21.9(78.1↓)	6.0(27.1↓)	19.8(80.2↓)
	L4	0.1(22.3↓)	—	0.0(82.9↓)	—	1.6(17.5↓)	—	4.8(47.5↓)	—	1.0(32.1↓)	—
Sonnet 4.6	L0	36.2	100.0	82.6	100.0	65.6	100.0	1.0 [†]	100.0	34.1	100.0
	L1	41.2(5.0↑)	61.0(39.0↓)	77.0(5.6↓)	44.5(55.5↓)	29.2(36.4↓)	43.5(56.5↓)	6.0(5.0↑)	30.0(70.0↓)	18.5(15.6↓)	44.1(55.9↓)
	L2	25.5(10.6↓)	36.2(63.8↓)	38.7(43.9↓)	24.6(75.4↓)	8.8(56.7↓)	26.3(73.7↓)	3.9(2.9↑)	16.9(83.1↓)	4.6(29.5↓)	29.4(70.6↓)
	L3	9.9(26.2↓)	27.0(73.0↓)	5.5(77.1↓)	19.1(80.9↓)	1.9(63.7↓)	18.6(81.4↓)	1.7(0.7↑)	12.1(87.9↓)	2.1(31.9↓)	22.6(77.4↓)
	L4	0.4(35.8↓)	—	0.0(82.5↓)	—	1.1(64.5↓)	—	0.7(0.3↓)	—	0.6(33.5↓)	—
DeepSeek-R1	L0	41.9	100.0	30.0	100.0	43.4	100.0	33.3	100.0	20.9	100.0
	L1	20.7(21.2↓)	79.5(20.5↓)	14.4(15.6↓)	12.7(87.3↓)	18.4(25.1↓)	18.8(81.2↓)	14.8(18.5↓)	35.4(64.6↓)	11.5(9.4↓)	23.2(76.8↓)
	L2	6.2(35.7↓)	54.9(45.1↓)	2.7(27.3↓)	9.6(90.4↓)	9.2(34.3↓)	13.8(86.2↓)	7.7(25.6↓)	33.5(66.5↓)	6.3(14.6↓)	12.9(87.1↓)
	L3	2.5(39.4↓)	48.1(51.9↓)	0.5(29.5↓)	11.3(88.7↓)	6.0(37.5↓)	14.1(85.9↓)	4.9(28.4↓)	32.6(67.4↓)	4.8(16.1↓)	10.9(89.1↓)
	L4	0.2(41.7↓)	—	0.0(30.0↓)	—	1.1(42.3↓)	—	1.6(31.7↓)	—	0.7(20.2↓)	—
Gemma-4-E4B	L0	65.1	100.0	71.2	100.0	6.0 [†]	100.0	4.4 [†]	100.0	5.4 [†]	100.0
	L1	42.2(22.9↓)	67.0(33.0↓)	55.6(15.6↓)	38.0(62.0↓)	5.4(0.6↓)	33.3(66.7↓)	5.1(0.7↑)	47.2(52.8↓)	4.0(1.4↓)	37.1(62.9↓)
	L2	7.7(57.4↓)	26.4(73.6↓)	16.8(54.4↓)	15.4(84.6↓)	2.4(3.7↓)	17.7(82.3↓)	2.1(2.3↓)	29.4(70.6↓)	1.6(3.8↓)	14.5(85.5↓)
	L3	2.2(62.9↓)	12.6(87.4↓)	2.5(68.7↓)	8.2(91.8↓)	1.6(4.5↓)	9.3(90.7↓)	1.2(3.2↓)	17.4(82.6↓)	1.2(4.1↓)	8.0(92.0↓)
	L4	0.1(65.0↓)	—	0.1(71.1↓)	—	1.0(5.0↓)	—	2.1(2.3↓)	—	0.5(4.9↓)	—

Table 10: Per-level task accuracy (%) and bidirectional NLI entailment rate (%) under Condition A (input compression) for all eight evaluated models. NLI % is the rate at which a generation entails its L0 counterpart bidirectionally; L0 is 100 % by construction. Colored subscripts show the change relative to each model’s L0 baseline (red ↓ for degradation, green ↑ for improvement). L4 NLI not reported; see §3.3.

[†] L0 extraction rate < 0.95; affected cells are audited in Appendix G.

Model	Lv	GSM8K		BoolQ		ARC-Easy		CSQA		MMLU-STEM	
		Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑	Acc ↑	NLI ↑
Qwen2.5-VL-7B	L0	82.4	100.0	79.4	100.0	95.4	100.0	76.5	100.0	62.9 [†]	100.0
	L1	72.3(10.1↓)	64.5(35.5↓)	85.1(5.7↑)	24.6(75.4↓)	94.2(1.2↓)	38.4(61.6↓)	76.7(0.2↑)	44.4(55.6↓)	66.6(3.7↑)	28.5(71.5↓)
	L2	46.9(35.5↓)	44.4(55.6↓)	83.2(3.8↑)	21.8(78.2↓)	86.8(8.6↓)	36.3(63.7↓)	74.0(2.5↓)	42.5(57.5↓)	61.0(1.9↓)	24.5(75.5↓)
	L3	36.7(45.7↓)	34.1(65.9↓)	84.2(4.8↑)	27.8(72.2↓)	89.2(6.2↓)	23.7(76.3↓)	56.5(20.0↓)	24.3(75.7↓)	55.5(7.4↓)	16.7(83.3↓)
	L4	23.8(58.6↓)	—	85.0(5.6↑)	—	94.8(0.6↓)	—	78.2(1.7↑)	—	62.7(0.2↓)	—
Qwen3.5-9B	L0	21.5	100.0	62.0	100.0	19.9	100.0	11.5	100.0	11.7	100.0
	L1	11.4(10.1↓)	66.9(33.1↓)	37.8(24.2↓)	40.9(59.1↓)	12.8(7.1↓)	21.4(78.6↓)	8.7(2.8↓)	34.7(65.3↓)	5.9(5.8↓)	20.3(79.7↓)
	L2	2.7(18.8↓)	16.4(83.6↓)	37.8(24.2↓)	7.3(92.7↓)	5.0(14.9↓)	5.9(94.1↓)	6.4(5.1↓)	9.1(90.9↓)	3.5(8.2↓)	5.3(94.7↓)
	L3	2.0(19.5↓)	17.4(82.6↓)	37.8(24.2↓)	14.3(85.7↓)	1.8(18.1↓)	6.4(93.6↓)	0.7(10.8↓)	15.9(84.1↓)	1.2(10.5↓)	6.6(93.4↓)
	L4	1.1(20.4↓)	—	0.0(62.0↓)	—	0.0(19.9↓)	—	0.0(11.5↓)	—	0.0(11.7↓)	—
GPT-4o	L0	91.5	100.0	87.6	100.0	97.9	100.0	85.3	100.0	71.2 [†]	100.0
	L1	95.6(4.1↑)	80.1(19.9↓)	87.8(0.2↑)	36.2(63.8↓)	98.8(0.9↑)	37.2(62.8↓)	84.7(0.6↓)	57.4(42.6↓)	85.1(13.9↑)	30.7(69.3↓)
	L2	93.4(1.9↑)	80.5(19.5↓)	87.2(0.4↓)	40.5(59.5↓)	98.5(0.6↑)	34.9(65.1↓)	85.9(0.6↑)	57.5(42.5↓)	81.9(10.7↑)	32.5(67.5↓)
	L3	88.9(2.6↓)	78.8(21.2↓)	89.1(1.5↑)	34.4(65.6↓)	98.7(0.8↑)	35.1(64.9↓)	85.2(0.1↓)	55.4(44.6↓)	81.5(10.3↑)	31.5(68.5↓)
	L4	57.5(34.0↓)	—	90.8(3.2↑)	—	98.9(1.0↑)	—	85.8(0.5↑)	—	77.4(6.2↑)	—
GPT-5.4	L0	96.9	100.0	90.1	100.0	99.1	100.0	85.8	100.0	93.0	100.0
	L1	96.3(0.6↓)	65.4(34.6↓)	88.9(1.2↓)	35.8(64.2↓)	99.1	30.6(69.4↓)	84.8(1.0↓)	40.6(59.4↓)	93.2(0.2↑)	46.3(53.7↓)
	L2	96.7(0.2↓)	65.7(34.3↓)	87.3(2.8↓)	46.7(53.3↓)	99.2(0.1↑)	35.0(65.0↓)	85.4(0.4↓)	41.9(58.1↓)	91.8(1.3↓)	47.2(52.8↓)
	L3	95.9(1.0↓)	66.6(33.4↓)	87.9(2.2↓)	36.5(63.5↓)	98.8(0.3↓)	33.4(66.6↓)	85.8	42.0(58.0↓)	86.9(6.1↓)	46.4(53.6↓)
	L4	69.2(27.7↓)	—	90.7(0.6↑)	—	98.8(0.3↓)	—	83.2(2.6↓)	—	82.1(10.9↓)	—
Haiku 4.5	L0	95.1	100.0	85.7	100.0	96.9	100.0	83.7	100.0	65.8	100.0
	L1	95.9(0.8↑)	58.5(41.5↓)	86.2(0.5↑)	49.4(50.6↓)	97.1(0.2↑)	36.9(63.1↓)	82.7(1.0↓)	67.1(32.9↓)	72.6(6.8↑)	32.9(67.1↓)
	L2	90.8(4.3↓)	58.4(41.6↓)	84.2(1.5↓)	52.8(47.2↓)	81.5(15.4↓)	35.0(65.0↓)	77.5(6.2↓)	69.7(30.3↓)	43.5(22.3↓)	29.0(71.0↓)
	L3	76.7(18.4↓)	52.0(48.0↓)	83.9(1.8↓)	51.4(48.6↓)	48.9(48.0↓)	28.7(71.3↓)	57.9(25.8↓)	65.1(34.9↓)	25.7(40.1↓)	21.8(78.2↓)
	L4	12.7(82.4↓)	—	87.5(1.8↑)	—	95.2(1.7↓)	—	81.3(2.4↓)	—	41.7(24.1↓)	—
Sonnet 4.6	L0	97.1	100.0	89.7	100.0	98.7	100.0	88.2	100.0	80.4 [†]	100.0
	L1	97.4(0.3↑)	59.3(40.7↓)	90.0(0.3↑)	32.7(67.3↓)	98.7	20.7(79.3↓)	87.3(0.9↓)	42.2(57.8↓)	92.7(12.3↑)	33.5(66.5↓)
	L2	97.3(0.2↑)	57.8(42.2↓)	89.4(0.3↓)	37.6(62.4↓)	98.9(0.2↑)	23.5(76.5↓)	86.8(1.4↓)	48.9(51.1↓)	89.0(8.6↑)	33.9(66.1↓)
	L3	93.0(4.1↓)	59.6(40.4↓)	89.0(0.7↓)	36.2(63.8↓)	98.2(0.5↓)	29.3(70.7↓)	85.8(2.4↓)	53.1(46.9↓)	76.5(3.9↓)	34.2(65.8↓)
	L4	38.5(58.6↓)	—	90.9(1.2↑)	—	99.2(0.5↑)	—	87.4(0.8↓)	—	81.2(0.8↑)	—
DeepSeek-R1	L0	51.3	100.0	63.4	100.0	56.7	100.0	43.6	100.0	29.9	100.0
	L1	62.5(11.2↑)	74.4(25.6↓)	50.0(13.4↓)	38.4(61.6↓)	34.2(22.5↓)	29.4(70.6↓)	25.6(18.0↓)	47.2(52.8↓)	19.1(10.8↓)	27.9(72.1↓)
	L2	23.8(27.4↓)	69.0(31.0↓)	32.4(31.0↓)	38.1(61.9↓)	26.5(30.2↓)	31.1(68.9↓)	18.3(25.3↓)	51.1(48.9↓)	12.3(17.6↓)	28.2(71.8↓)
	L3	5.0(46.2↓)	64.3(35.7↓)	13.0(50.4↓)	37.0(63.0↓)	9.8(46.9↓)	29.5(70.5↓)	7.1(36.4↓)	52.7(47.3↓)	5.1(24.8↓)	25.0(75.0↓)
	L4	0.3(50.9↓)	—	0.1(63.3↓)	—	0.6(56.1↓)	—	0.7(42.9↓)	—	0.4(29.5↓)	—
Gemma-4-E4B	L0	74.0	100.0	82.8	100.0	84.3	100.0	71.7	100.0	36.5 [†]	100.0
	L1	88.6(14.6↑)	54.2(45.8↓)	84.0(1.2↑)	46.4(53.6↓)	95.8(11.5↑)	28.6(71.4↓)	75.1(3.4↑)	54.7(45.3↓)	71.9(35.3↑)	19.9(80.1↓)
	L2	86.3(12.3↑)	51.9(48.1↓)	84.9(2.1↑)	43.9(56.1↓)	95.7(11.4↑)	25.4(74.6↓)	75.4(3.8↑)	50.4(49.6↓)	70.2(33.7↑)	15.5(84.5↓)
	L3	72.1(1.9↓)	47.7(52.3↓)	84.0(1.2↑)	39.8(60.2↓)	94.3(10.0↑)	28.5(71.5↓)	72.8(1.1↑)	51.4(48.6↓)	62.1(25.6↑)	16.8(83.2↓)
	L4	20.6(53.4↓)	—	84.6(1.8↑)	—	96.2(11.9↑)	—	75.8(4.1↑)	—	64.6(28.1↑)	—

Table 11: Per-level task accuracy (%) and bidirectional NLI entailment rate (%) under Condition B (output constraint) for all eight evaluated models. NLI % is the rate at which a generation entails its L0 counterpart bidirectionally; L0 is 100 % by construction. Colored subscripts show the change relative to each model’s L0 baseline (red ↓ for degradation, green ↑ for improvement). The contrast between green Acc deltas and red NLI deltas on classification benchmarks under non-reasoning models is the surface-text divergence of Finding 2. L4 NLI not reported; see §3.3.

[†] L0 extraction rate < 0.95; affected cells are audited in Appendix G.

Condition A (input) Condition B (output)



Figure 5: Per-level accuracy across the eight evaluated models and five benchmarks under both conditions. Solid: Condition A. Dashed: Condition B. The per-level bidirectional NLI grid is released alongside the artifact bundle.

Benchmark	Model	Cond A		Cond B	
		L_c^{acc}	L_c^{sem}	L_c^{acc}	L_c^{sem}
GSM8K	Qwen2.5-VL-7B	L1	L1	L1	L1
	GPT-4o	L1	L1	L4	L1
	Haiku 4.5	L2	L1	L3	L1
BoolQ	Qwen2.5-VL-7B	L1	L1	—	L1
	GPT-4o	L1	L1	—	L1
	Haiku 4.5	L1	L1	—	L1
ARC-Easy	Qwen2.5-VL-7B	L1	L1	L2	L1
	GPT-4o	L1	L1	—	L1
	Haiku 4.5	L2	L1	L2	L1
CommonsenseQA	Qwen2.5-VL-7B	L1	L1	L3	L1
	GPT-4o	L1	L1	—	L1
	Haiku 4.5	L1	L1	L2	L1
MMLU-STEM	Qwen2.5-VL-7B	L1	L1	L3	L1
	GPT-4o	L2	L1	—	L1
	Haiku 4.5	L1	L1	L2	L1

Table 12: Threshold levels at which accuracy ($L_c^{\text{acc}}, \geq 5$ pp below L0, scanned L1–L4) and NLI ($L_c^{\text{sem}}, \geq 15$ pp below L0, scanned L1–L3 as L4 has no NLI) first cross the degradation criterion; “—” marks a cell where the threshold is never crossed in that range. Of 80 ‘(model, dataset, condition)’ cells, 60 are evaluable (both thresholds crossed in range), and $L_c^{\text{sem}} \leq L_c^{\text{acc}}$ on all 60 on the full eight-model panel under the strict bidirectional NLI criterion; the rows shown are a representative three-model excerpt. The frequent “—” under Cond B reflects accuracy that never degrades by 5 pp under output compression (Finding 2).

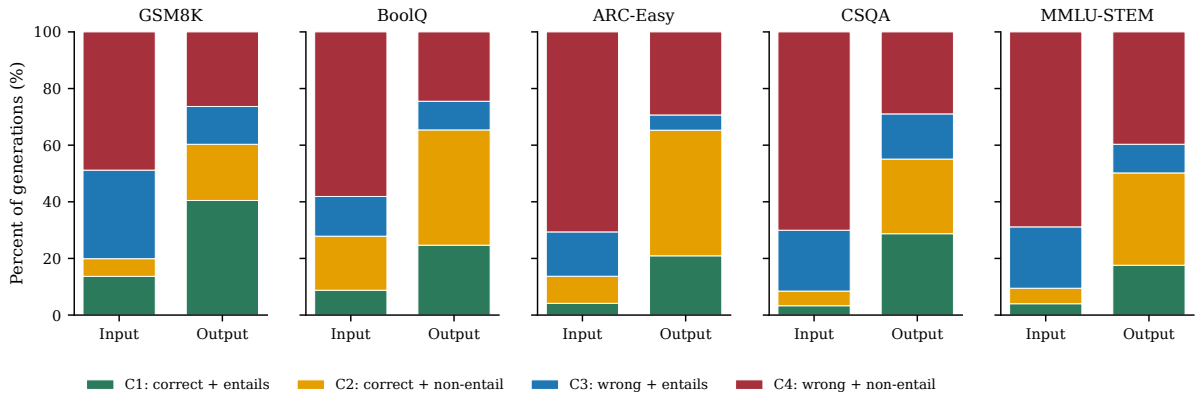


Figure 6: 2×2 dissociation by dataset, aggregated across L1–L3 and the eight evaluated models. Each panel shows Condition A and Condition B side by side; bar segments are the C_1 – C_4 outcome shares.

Benchmark	Cond A	Cond B
GSM8K	32.6%	34.6%
BoolQ	64.8%	61.9%
ARC-Easy	68.1%	69.6%
CommonsenseQA	66.0%	51.5%
MMLU-STEM	61.8%	70.0%

Table 13: Mean L1 semantic non-preservation (%) by benchmark under the headline bidirectional NLI criterion, averaged across the eight evaluated models. The same task-type ordering holds under the alternative criteria of §3.3 (Appendix C).

C Judge Reliability and Semantic Robustness

Calibration and robustness evidence for Finding 2: judge reliability by compression level (§C.1, Table 14) and cross-metric replication under eleven additional measures (Table 15). The headline NLI rate sits near the conservative end of the family.

C.1 NLI Judge Reliability by Compression Level

Calibration uses 70 POS-filtered synthetic positive pairs per level: pairs are semantically equivalent by construction, so non-entailment is attributable to the judge. The Disagree-A rate (NLI fail with cosine > 0.85) at L1 Cond B is 17.1%, leaving judge-failure cases a clear minority. L1 is judge-reliable (FN 2.9%), L2 supplementary (FN 28.6%), L3 descriptive only (FN 50.0%); L4 is excluded.

C.2 Robustness Across Alternative Semantic Measures

We re-scored every (compressed, L0) pair under the eleven complementary C_2 measures that, with the headline, make up the twelve: forward-only and soft NLI, three independent NLI judges (BART-large-MNLI, multilingual XNLI, DeBERTa-large), a faithfulness checker (MiniCheck), learned and surface similarity (BLEURT, ROUGE-L, METEOR), an STS cross-encoder, and a QA-based propositional check. Continuous-similarity scores (BERTScore, two sentence-embedding cosines) and an answer-anchored NLI variant are reported alongside for context. Table 15 reports L1 Cond B aggregates on the 6-non-reasoning panel. All twelve C_2 measures report a substantial divergence on the 6-non-reasoning panel (roughly 41–88%), with the headline bidirectional NLI near the conservative end of the family. Per-metric details and the full per-cell breakdown are released in our repository.

C.3 Length-Controlled NLI Re-Scoring

Bounds the length-and-register confound. For each (L0, L1-B) pair we truncate L0 to the L1-B wordpiece-token length and re-score with the same judge under the same bidirectional criterion.

Procedure. For each (model, dataset, item) tuple at L1-B, tokenize L0, truncate to $\min(|L0|, |L1-B|)$, detokenize, re-score in both directions. Same denominator as the Finding 2 headline.

Headline. The C_2 rate *rises* by +28.4 pp on the 6-non-reasoning panel and +21.6 pp on the 8-model aggregate under length-matched scoring (Table 16). Reasoning models (DeepSeek-R1, Qwen3.5-9B) move by ≤ 7 pp because their L0 outputs are already short.

Per-cell pattern. Same direction on every non-reasoning ‘(model, dataset)’ cell. Largest shifts: GPT-4o on GSM8K (18.3% \rightarrow 92.0%, +73.8 pp), Sonnet on GSM8K (+55.8 pp), Qwen2.5-VL-7B on GSM8K (+55.0 pp).

System-prompt noise floor. We do not separately score an L0-A vs. L0-B paired-NLI baseline that would isolate the system-prompt register shift between conditions; the headline should be read as a floor above any such (unmeasured) noise.

Level	C_2 strict (grand)	C_2 fwd-only	Disagree-A (Cond B)	Judge FN (calib)	Tag
L1	42.7 %	38.8 %	17.1 %	2.9 %	Robust (Cond B)
L2	39.5 %	34.8 %	14.1 %	28.6 %	Uncertain
L3	36.7 %	30.2 %	8.5 %	50.0 %	Unreliable
L4			excluded from semantic evaluation		

Table 14: NLI judge reliability by compression level (8-model grand aggregate, Cond B). C_2 strict: bidirectional entailment. C_2 fwd-only: forward direction only. Disagree-A: NLI fails but cosine > 0.85 (Cond B). Judge FN: false-negative rate on POS-filtered synthetic positive pairs. L1’s 2.9% FN underpins the L1 headline.

Measure	L1 Cond B value	Type
<i>The twelve C_2 measures (thresholded)</i>		
NLI bidirectional	51.9 % C_2	Thresholded rate
NLI forward-only	47.4 % C_2	Thresholded rate
NLI soft (mean prob > 0.5)	41.2 % C_2	Thresholded rate
BART-large-MNLI bidirectional	56.1 % C_2	Independent NLI judge
mDeBERTa-v3-XNLI bidirectional	50.2 % C_2	Multilingual NLI judge
MiniCheck	54.0 % C_2	Faithfulness checker
DeBERTa-large bidirectional	49.5 % C_2	Larger NLI variant
BLEURT (Elron/bleurt-base-128)	87.6 % C_2 (BLEURT < 0)	Learned similarity
METEOR	81.9 % C_2 (METEOR < 0.3)	Paraphrase-aware surface
ROUGE-L	55.9 % C_2 (ROUGE-L < 0.3)	Surface overlap
STS cross-encoder	87.2 % C_2 (STS-pass < 0.5)	STS
QA-based propositional score	60.5 % C_2	Content recovery
<i>Reported for context (not counted)</i>		
BERTScore (roberta-large)	0.858 mean F1	Token-level
Cosine (MiniLM, paper)	0.725 mean cosine	Embedding
Cosine (intfloat/e5-base-v2)	0.904 mean cosine	Embedding, 2nd arch.
Answer-anchored NLI	11.4 % C_2	Ground truth
Judge calibration (FN on L1 positives)	2.9 %	Lower bound

Table 15: Cross-metric agreement at L1 Condition B (6-non-reasoning panel). The twelve C_2 measures all report a substantial divergence on the 6-non-reasoning panel (41%–88%); the headline bidirectional NLI (51.9% on the 6-non-reasoning panel) sits near the conservative end of the family, which ranges from DeBERTa-large (49.5% on the 6-non-reasoning panel) to BLEURT (87.6% on the 6-non-reasoning panel). Continuous-similarity scores (BERTScore, cosine) and the answer-anchored check are shown for context. Answer-anchored NLI is lower because a correct answer usually entails the ground-truth-answer hypothesis even when the surrounding reasoning text diverges.

Panel	Original C_2	Length-controlled C_2	Δ
6 non-reasoning models (headline)	51.9 %	80.4 %	+28.4 pp
8 models (grand aggregate)	42.7 %	64.2 %	+21.6 pp

Table 16: Length-controlled bidirectional NLI C_2 at L1 Condition B. L0 truncated to L1-B’s wordpiece-token length; same judge, same denominator as the headline.

D Comparison with LLMingua-2

LLMingua-2 was designed for compressing few-shot demonstrations and long-context inputs, not individual questions in short benchmarks. Our comparison applies it outside its intended regime; we report it because it is the standard learned input-side baseline. We re-ran the input channel under LLMingua-2 (Pan et al., 2024) on 3 models (GPT-4o, Sonnet 4.6, Qwen2.5-VL-7B) \times 3 datasets (GSM8K, BoolQ, ARC-Easy) at the paper-default $\tau=0.5$ rate and at $\tau=0.8$ on Qwen2.5-VL-7B (matched to CAVEWOMAN’s telegraphic retention). Table 17 and Figure 7 report cross-cell aggregates.

Method	Mean acc.	Mean NLI	Mean C_2
CAVEWOMAN (telegraphic)	0.53	0.46	0.27
CAVEWOMAN (keyword)	0.22	0.27	0.14
LLMingua-2 ($\tau = 0.5$)	0.15	0.25	0.08
LLMingua-2 ($\tau = 0.8$) [†]	0.45	0.38	0.22

Table 17: Input compression, 3 models \times 3 datasets at $\tau=0.5$; $\tau=0.8$ was run on Qwen2.5-VL-7B only as a rate-sensitivity check at a retention rate matched to CAVEWOMAN’s telegraphic level. $C_2 > 0$ on every cell.

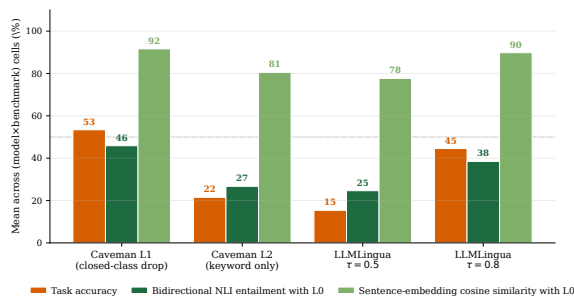


Figure 7: Mean accuracy, bidirectional NLI entailment against L0, and dissociation rate (C_2) for four input-compression configurations, averaged over the 3 models \times 3 datasets the comparison was run on ($\tau=0.8$ is Qwen2.5-VL-7B only; see Table 17). $C_2 > 0$ under every method; the divergence reproduces beyond the POS filter.

Three results. (i) The divergence generalizes beyond our POS filter: $C_2 > 0$ on every cell under both methods. (ii) LLMingua-2’s published

GSM8K robustness does not transfer to question-text compression: GSM8K accuracy at the default rate is 20–35% across the three models, well below the $\sim 79\%$ they report for few-shot demonstration compression. (iii) The collapse is rate-driven, not method-driven: at $\tau=0.8$, LLMingua-2 on Qwen2.5-VL-7B GSM8K reaches 0.66 accuracy and 0.73 NLI rate, comparable to CAVEWOMAN’s telegraphic level (the lower $\tau=0.8$ values in Table 17 are the three-dataset Qwen2.5-VL-7B mean).

Compressed prompt at $\tau=0.5$. Same GSM8K question (Janet’s ducks, ground-truth answer 18) before and after LLMingua-2 compression. Articles, prepositions, and “how much” phrasing are removed; the model often fails. The $\tau=0.8$ check preserves 51 of 64 tokens and recovers most of the accuracy gap; per-cell numbers are released in our repository.

Original prompt (64 tokens)

Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?

LLMingua-2 at $\tau=0.5$ (30 tokens; $\approx 47\%$ kept)

Janet ducks lay 16 eggs. eats three breakfast bakes muffins. sells remainder farmers’ market \$ 2 fresh duck egg. dollars?

Structural failure mode at the default rate. LLMingua-2 prunes single-letter MCQ labels (“A:”, “B:”, ...) as low-information tokens, collapsing ARC-Easy accuracy to 6–7% across all three models at $\tau=0.5$. CAVEWOMAN’s POS filter retains these tokens by construction.

E Reasoning-Token Accounting (DeepSeek-R1 and GPT-5.4)

DeepSeek-R1-Distill-Qwen-7B emits hidden `<think>...</think>` traces before its visible answer; HuggingFace generate counts these against `max_new_tokens`, so the per-level decoder budget covers the trace and the visible response together. The trace counts as `output_tokens` and enters the NLI judge when it survives. At L4 (20-subword cap) the budget is exhausted

before the visible answer starts, which is why DeepSeek-R1 collapses to near-zero accuracy at L4 on every dataset (Table 11); at L1–L3, partial traces bias bidirectional NLI against DeepSeek-R1 relative to non-reasoning models.

The pattern also inverts DeepSeek-R1’s accuracy-vs-reference-text cells: C_2 is small ($\sim 19\%$) on the DeepSeek-R1 L1 panel and C_3 (agreement despite incorrect) is comparatively large ($\sim 21\%$) there. The reported output/input accuracy ratio of 2.4 is therefore not apples-to-apples with the non-reasoning panel, since the visible budget is effectively shorter.

GPT-5.4 exhibits the same accounting on the API side. Its billed output_tokens include server-side reasoning tokens that never appear in the returned text: at L0 Condition B the visible response averages ~ 56 tokens on BoolQ while the billed output implies ~ 198 , a $3.5\times$ gap on that benchmark ($2.9\times$ averaged across the five benchmarks); GPT-4o, Haiku, and Sonnet bill exactly their visible output. Because the visible-output constraint shortens only the returned text and not the hidden reasoning, output compression leaves GPT-5.4’s priced token count essentially unchanged, while the L1 telegraphic system prompt adds ~ 105 input tokens; its realized cost is therefore flat to slightly higher under output compression (Finding 1). Unlike DeepSeek-R1, these reasoning tokens are not part of the visible text and so do not enter the bidirectional NLI judge, so GPT-5.4’s surface-text divergence (Finding 2) is measured on visible text alone and is unaffected.

A third reasoning-protocol model, Kimi-K2.6, was instrumented in the evaluation but excluded from the eight-model panel for a related reason: its constrained-output responses (L1 Condition B and stricter) returned empty visible text on 99–100% of items, with token consumption equal to the per-level max_new_tokens cap. The reasoning-block protocol consumes the entire budget under output constraint, the dual of the DeepSeek-R1 case: where DeepSeek-R1’s trace partially survives and biases entailment, Kimi’s consumes everything and leaves nothing visible. We note it because the failure mode marks a boundary of the output-constraint protocol for reasoning-block models.

F Limitations in Detail

Length and register confound. L1–L3 output-compression generations are forced into a shorter,

function-word-light register than L0, and the DeBERTa-NLI training distribution does not cover (telegraphic, prose) pairs. The length-controlled re-scoring of Appendix C.3 bounds this; BART-MNLI (60.3% inter-judge agreement, 45.7% both-fail at L1-B) and answer-anchored NLI corroborate.

Judge reliability above L1. See Appendix C.1. L1 is judge-reliable; L2 is supplementary; L3 is descriptive only. L4 is excluded.

Reference noise and multiple-comparisons correction. The divergence is measured against one greedy L0 draw; sampled-decoding stochasticity is not measured. The L0-A vs. L0-B noise floor is not separately measured. Across the 320-cell Wilcoxon family at $\alpha = 0.05$, 278 of 320 cells are significant uncorrected and 278 remain after Benjamini–Hochberg correction (zero lost). Per-cell bootstrap CIs ($n_{\text{boot}} = 1,000$) span 1–7 pp.

G Extraction-Rate Audit and L4 Length Distribution

G.1 Extraction-Rate Audit

Accuracy is the fraction of items whose extracted answer matches the ground-truth answer; unextracted items count as incorrect. When L0 parse rate is materially below 1.0, “L1 exceeds L0” partially reflects the extractor working better on shorter outputs.

Headline gap on MMLU-STEM (Condition B). Among the models showing apparent L1 gains, L0-B MMLU-STEM parse rate ranges from 0.492 (Gemma-4-E4B) to 0.854 (Sonnet) and recovers to 0.807–0.969 at L1-B. Largest parse-rate gaps: Gemma-4-E4B (+38.7 pp parse rate), GPT-4o (+16.8 pp parse rate), Qwen2.5-VL-7B (+14.1 pp parse rate). We do not report L1-vs-L0 accuracy gains where L0 parse < 0.95 .

Other affected cells. 45 of 80 (model, dataset, condition)’ cells have L0 parse < 0.95 . Sonnet L0-A CSQA is the most extreme (0.037) and drives the apparent “Sonnet L1-A exceeds L0-A by 5 pp on CSQA” artifact in Table 10. Qwen3.5-9B’s below-random L0-B accuracies on ARC-Easy, CommonsenseQA, and MMLU-STEM are not explained by the extraction-rate evidence reported here. The full per-cell audit is released in our repository.

G.2 L4 Output-Length Distribution

L4’s 15-token target is conveyed through the prompt with a `max_new_tokens = 20` ceiling (Table 18).

Metric	L4-A (budget 15, soft)	L4-B (budget 20)
<i>n</i> items	91,720	91,720
Weighted mean output tokens	19.8	9.9
Weighted violation rate (> 15 A, > 20 B)	99.0%	0.0%

Table 18: L4 output-length distribution (40 model×dataset cells per condition). Cond A overshoots the 15-token target; the same ceiling under Cond B binds tightly.

L4-A is retained in per-level tables with the soft-constraint caveat: “asked to compress to 15 tokens but allowed up to 20” rather than a strict 15-token budget.

H Constraint-Level Specifications

Verbatim system prompts and POS-tag filter rules for both conditions; the implementing code is released in our repository.

H.1 Condition A: Input-Compression Filter

System prompt is fixed across all five levels; only the user message changes via a deterministic spaCy POS-tag filter. Surviving tokens are rejoined with single whitespace; an empty filter output falls back to the original text.

Neutral system prompt (identical at all levels).

Condition A: System Prompt (L0–L4)

You are a helpful assistant. Answer the following question accurately and completely.

POS-tag filter rules. Table 19 gives the per-level filter rule applied to the user message. The L4 row truncates the *token stream* (not the character sequence), preserving the monotone ladder $L_0 \supseteq L_1 \supseteq L_2 \supseteq L_3 \supseteq L_4$.

H.2 Condition B: Per-Level System Prompts

Each prompt has a named constraint type, a rule list, a task-neutral example, and the answer-format convention. Decoder budgets: $\{L_0:400, L_1:300, L_2:200, L_3:150, L_4:20\}$ `max_new_tokens`.

Condition B: L0 (Unconstrained)

Answer the following question accurately. Reason step by step in full, grammatical English sentences. Conclude with the final answer on its own line in the form 'Answer: <answer>'.

The final-line answer matches what the question asks for: a number for numeric questions, 'yes' or 'no' for yes/no questions, or a single letter (A, B, C, ...) for multiple-choice questions.

Condition B: L1 (Telegraphic)

Answer the question under a TELEGRAPHIC constraint.

Rules:

- DO NOT use any function words. No articles (the, a, an). No conjunctions (and, but, or, so). No prepositions (of, in, to, for, at, with, from, by, on, per).
- DO use nouns, main verbs, numbers, and standard symbols (+, -, *, /, =).
- Show each reasoning step.
- End with a line: 'Answer: <answer>'.

Example (task-neutral):

Premise mentions item X. Property Y holds X. Match: yes.
Answer: <answer>

The final-line answer matches what the question asks for: a number for numeric questions, 'yes' or 'no' for yes/no questions, or a single letter (A, B, C, ...) for multiple-choice questions.

Level	Rule
L0	unchanged (original text)
L1	drop DT, IN, CC, RP, TO, MD
L2	keep NN, NNS, NNP, NNPS, VB, VBZ, VBD, VBN, VBG, CD
L3	keep NN, NNS, NNP, NNPS, CD
L4	apply L3 filter, then keep first 15 whitespace tokens

Table 19: Condition A per-level POS-tag filter rules.

L0	Janet's ducks lay 16 eggs per day . She eats three for breakfast every morning and bakes muffins for her friends every day with four . She sells the remainder at the farmers' market daily for \$ 2 per fresh duck egg . How much in dollars does she make every day at the farmers' market ?
L1	Janet's ducks lay 16 eggs per day . She eats three for breakfast <i>every morning</i> and bakes muffins for <i>her friends every</i> day with four . She sells the remainder at the farmers' market <i>daily</i> for \$ 2 per <i>fresh duck</i> egg . How much in dollars does she make <i>every</i> day at the farmers' market ?
L2	Janet's ducks <i>lay</i> 16 eggs <i>per</i> day . <i>She eats three for</i> breakfast <i>every morning and bakes</i> muffins <i>for her friends every</i> day <i>with</i> four . <i>She sells the</i> remainder <i>at the</i> farmers' market <i>daily for</i> \$ 2 <i>per fresh duck</i> egg . <i>How much in</i> dollars <i>does she make</i> every day <i>at the</i> farmers' market ?
L3	Janet's ducks <i>lay</i> 16 eggs <i>per</i> day . <i>She eats three for</i> breakfast <i>every morning and bakes</i> muffins <i>for her friends every</i> day <i>with</i> four . <i>She sells the</i> remainder <i>at the</i> farmers' market <i>daily for</i> \$ 2 <i>per fresh duck</i> egg . <i>How much in</i> dollars <i>does she make</i> every day <i>at the</i> farmers' market ?
L4	Janet's <i>ducks lay</i> 16 eggs <i>per day</i> . <i>She eats three for breakfast</i> every morning and bakes <i>muffins for her friends every</i> day <i>with</i> four . <i>She sells the remainder at the farmers' market</i> daily for \$ 2 <i>per fresh duck</i> egg . <i>How much in</i> dollars <i>does she make</i> every <i>day at the farmers' market</i> ?

Figure 8: Worked example of the input-compression filter applied to a single question at each of the five reduction levels. L0 is the unmodified prompt; L1 removes closed-class function words; L2 retains nouns, verbs, and cardinal numerals; L3 strips the verbs to leave a nominal skeleton; L4 truncates the L3 form to its first fifteen tokens. The same filter family defines both channels.

Condition B: L2 (Keyword-Only)

Answer the question under a KEYWORD-ONLY constraint.

Rules:

- Use ONLY nouns and main verbs. No grammar, no full sentences.
- Output as fragments, short labels, or list items.
- Numbers and standard symbols (+, -, *, /, =) are allowed.
- Each reasoning step appears as a fragment.
- End with a line: 'Answer: <answer>'.

Example:

Item: X
Property Y: holds
Match: yes
Answer: <answer>

The final-line answer matches what the question asks for: a number for numeric questions, 'yes' or 'no' for yes/no questions, or a single letter (A, B, C, ...) for multiple-choice questions.

Condition B: L3 (Noun-Phrase Skeleton)

Answer the question under a NOUN-PHRASE SKELETON constraint.

Rules:

- NO verbs of any kind. None.
- Use only nominal fragments: nouns, noun compounds, numbers, and standard symbols (+, -, *, /, =).
- Each step is a noun phrase labeling a quantity, claim, or property.
- End with a line: 'Answer: <answer>'.

Example:

Item: X
Property in question: Y
Match status: positive
Answer: <answer>

The final-line answer matches what the question asks for: a number for numeric questions, 'yes' or 'no' for yes/no questions, or a single letter (A, B, C, ...) for multiple-choice questions.

Condition B: L4 (Hard 15-Token Budget)

Answer the question under a HARD TOKEN BUDGET.

Rules:

- Your ENTIRE response must be 15 tokens or fewer.
- The response MUST include the final answer.
- Prefer the raw answer over prose.

Example: 'Answer: <answer>'

The final-line answer matches what the question asks for: a number for numeric questions, 'yes' or 'no' for yes/no questions, or a single letter (A, B, C, ...) for multiple-choice questions.

I Qualitative Examples

One BoolQ item (Qwen2.5-VL-7B, Condition B) traced across all five levels (L4 is shown for illustration only; it is excluded from the aggregate semantic scoring of §3.3). Box color: green = C_1 (correct, entailment PASS), orange = C_2 (correct, entailment FAIL), gray = L0. The correct answer survives compression after the reasoning chain has collapsed (the C_2 pattern of §4.2). A GSM8K arithmetic trace is in the released artifact bundle.

BoolQ: Passage Yes/No (Qwen2.5-VL-7B, Condition B)

Question: is harry potter and the escape from gringotts a roller coaster ride

Ground truth answer: yes

(BoolQ provides a supporting passage with each question; passage omitted here for space.)

L0 (Unconstrained baseline) Pred: yes GT: yes Correct

To determine if this attraction is a roller coaster ride, I analyzed the passage. The passage states it is an “indoor steel roller coaster” at Universal Studios Florida, which directly confirms it is a roller coaster. Based on this evidence. . .

Answer: yes

L1 (Telegraphic) Pred: yes GT: yes Correct
NLI: FAIL [C_2]

Yes.

Answer: yes

L2 (Keyword-Only) Pred: yes GT: yes Correct NLI: FAIL [C_2]
Yes

L3 (Noun-Phrase Skeleton) Pred: yes GT: yes Correct NLI: PASS [C_1]
Item: Harry Potter and the Escape from Gringotts Property in question: Roller coaster ride Match status: positive Answer: yes

L4 (Hard 15-Token Budget) Pred: yes GT: yes Correct NLI: FAIL [C_2]
Answer: Yes

Takeaway. The correct answer survives at every level, but the reasoning chain collapses at L1, L2, and L4 (NLI FAIL; C_2 cell). The L0 output makes several inferential steps citing the passage; “Yes.” and “Yes” make no such claims, so the L0 output does not entail them. L3 unexpectedly recovers bidirectional entailment: the noun-phrase skeleton labels the entity and property correctly, replicating enough of the L0 propositional structure for entailment to pass. This item exemplifies why accuracy alone cannot distinguish C_1 from C_2 .

J Released Artefacts and Reproducibility

Each (model, dataset, condition, level) configuration releases three artifacts in our repository: a base inference record (token counts, realized cost, extracted answer, ground-truth answer), a paired entailment record (bidirectional NLI scores at L1–L4), and a paired embedding record (sentence-embedding cosine at L0–L4). Each configuration is accompanied by a run-configuration manifest recording the git SHA, conda environment, package versions, and GPU. Aggregate per-cell summaries and a verification script that reproduces the paper’s reported numbers from those summaries are released alongside.