

Supplementary Material

CogniLoad: A Synthetic Natural Language Reasoning Benchmark With Tunable Length, Intrinsic Difficulty, and Distractor Density

The supplementary material is organized as follows:

- Appendix A investigates ρ relative to N and d .
- Appendix B analyses and discusses interaction effects among d , N , and ρ
- Appendix C charts model performance reported with error-bars.
- Appendix D discusses the evaluation, scoring pipeline, and error taxonomy.
- Appendix E reports the distribution of error types across models and different levels of d and N .
- Appendix F compares the AIC values of linear vs quadratic ρ in the GLM regression.
- Appendix G details the complete attribute ontology.
- Appendix H lists the token-length distribution of prompts and output models per model by d and N .

A Investigating ρ relative to N and d

In Figure 1 of the paper we discover a characteristic U-shape of the performance of LLMs on CogniLoad relative to ρ (i.e., the ratio of distractors to essential elements). The following figures investigate this U-shape at different levels of difficulty (Figure 1) and statement length (Figure 2).

Figure 1 plots mean accuracy as a function of the needle-to-hay ratio ρ (higher ρ means more signal per distractor) under three difficulty settings $d = 1, 5, 10$. For the easiest tasks ($d = 1$, left) accuracies are high and largely flat; most models show at most a shallow U-shape, with a mild dip around $\rho = 25\%–50\%$ and a small recovery by $\rho \geq 75\%$. At intermediate difficulty ($d = 5$, centre) the trough widens: many curves drop sharply from $\rho = 10\%$ to 25% and bottom out across $\rho \approx 25\%–50\%$ before rebounding at high ρ . A notable exception is EXAONE-Deep-32B, which trends downward as ρ increases for $d \geq 5$. Under the hardest condition ($d = 10$, right) capability tiers separate. Frontier models remain robust and still recover at high ρ —averaged over ρ , GPT-5 0,821, O3 0,804, GPT-5-mini 0,675, O4-mini 0,646, and Gemini-2.5-Pro 0,654—whereas mid-size and small models stay low despite a late uptick (e.g., DeepSeek-R1-0528 0,475; DeepSeek-R1-Distill-Llama-70B 0,361; QwQ-32B 0,300; Qwen3-32B 0,280; Phi-4-reasoning 0,256; Qwen3-1,7B 0,110). Overall, the panels show that (i) accuracy shifts downward and the U-shape becomes more pronounced as d increases; (ii) recovery at high ρ is strongly model-dependent and favours larger, reasoning-tuned systems; and (iii) the moderate- ρ regime ($\rho \approx 25\%–50\%$) remains the most adversarial, where balancing signal and noise is hardest.

Figure 2 shows mean accuracy as a function of the needle-to-hay ratio ρ for $N = 20$ (left), $N = 100$ (centre), and $N = 250$ (right). At $N = 20$ many systems still display a clear U-shape: accuracy begins near 1,0, dips at medium clutter around $\rho \approx 25\%–50\%$, and then rebounds as needles dominate. As the context grows, this dependency compresses and then largely disappears. At $N = 100$ the strongest models are nearly ρ -invariant (e.g., curves remain between about 0,93 and 0,97), while mid-tier models retain only a shallow trough. By $N = 250$ most curves are flat or gently increasing with ρ at much lower levels; only the frontier models exhibit a modest late uptick.

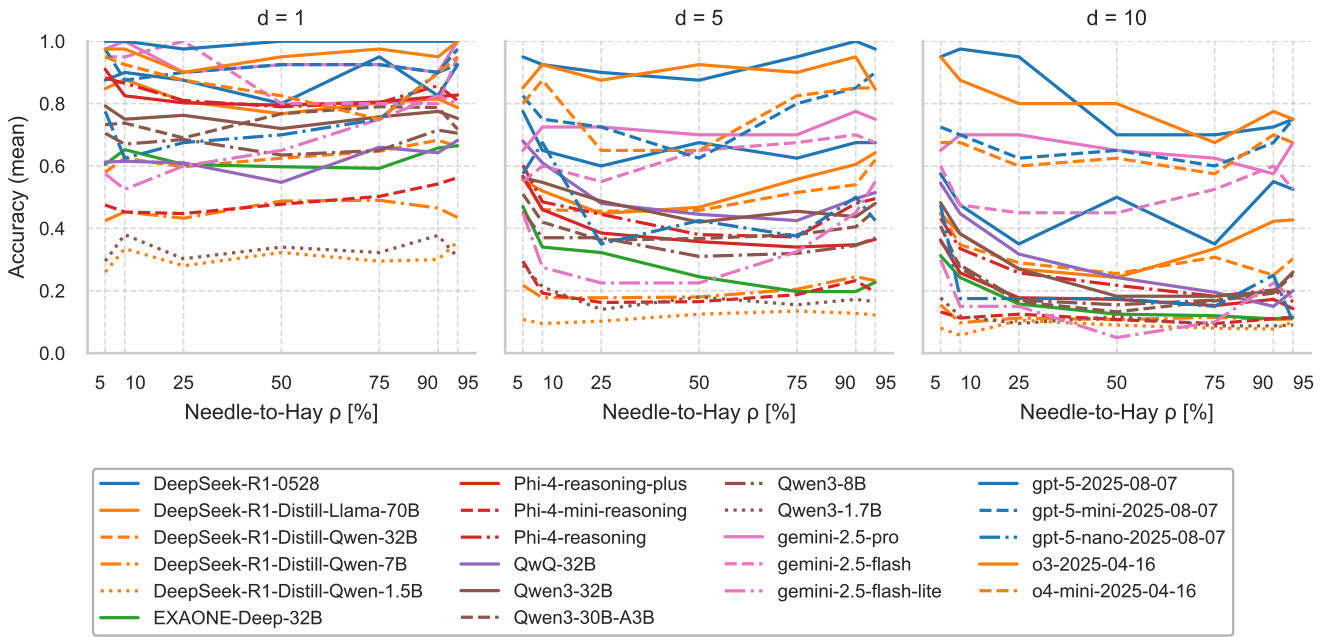


Figure 1: Distractor ratio ρ relative to d for $d \in \{1, 5, 10\}$

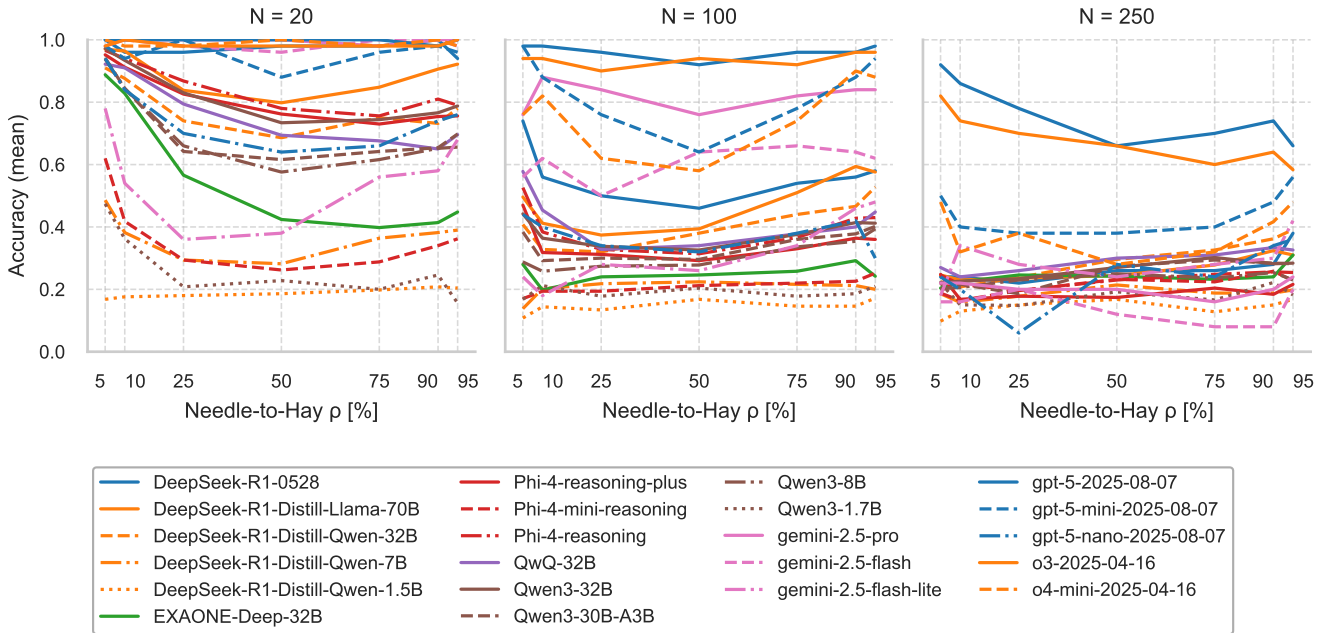


Figure 2: Distractor ratio ρ relative to N for $N \in \{20, 100, 250\}$

Taken together, the two figures show that difficulty d and length N both depress accuracy but modulate its dependence on ρ in different ways. Increasing d accentuates the classic U-shape (i.e., many models drop from

$\rho = 10\%$ into the mid-range $\rho \approx 25\%–50\%$ and then recover as needles dominate) whereas increasing N shifts the whole curve downward and progressively flattens it. These stresses reveal a clear stratification: frontier models remain comparatively robust (retaining some high- ρ gains and partial ρ -invariance), a mid-tier exhibits a pronounced but narrowing mid- ρ trough as N grows, and small models settle at low, nearly ρ -insensitive performance. In short, the mid- ρ band 25–50% is the most adversarial at fixed N and d , but at long contexts the primary failure mode becomes N itself rather than ρ .

B Interaction effects among d , N , and ρ

To probe whether load dimensions compound each other in predictable ways, we fit, for each model, a logistic GLM that augments the main-effects specification with all pairwise and three-way interactions among intrinsic difficulty d , effective length $\log_{10} N$, and the needle ratio ρ , while retaining a quadratic main effect for ρ to capture the observed U-shape:

$$\text{logit } P(\text{correct}) = \beta_0 + [d, \log_{10} N, \rho] + [d:\log_{10} N, d:\rho, \log_{10} N:\rho] + [d:\log_{10} N:\rho] + \rho^2.$$

In Table 1 we report for each interaction the coefficient β and an ANOVA-style likelihood-ratio p -value (p_{LR}) obtained by dropping that interaction from the full model (see Table).

B.1 Global patterns

Three robust patterns emerge across models:

- A strong $d \times N$ interaction is widespread and negative: 17/22 models show $p_{\text{LR}} < 0.01$ with $\beta_{d \times N} < 0$ (e.g., QwQ-32B: -0.39 , $p_{\text{LR}} = 1.25\text{e-}33$; Phi-4-reasoning: -0.45 , $p_{\text{LR}} = 1.94\text{e-}30$). This indicates that increases in intrinsic difficulty and sequence length compound to reduce accuracy additively. Frontier systems (gpt-5, o3) show no detectable $d \times N$ interaction (all $p_{\text{LR}} > 0.3$), consistent with their near-separable responses to d and N at current loads.
- A clear $d \times \rho$ interaction is present in 13/22 models (all $\beta_{d \times \rho} < 0$, $p_{\text{LR}} \leq 0.01$ in those cases; e.g., QwQ-32B: -0.96 , EXAONE-Deep-32B: -0.89). This confirms that higher intrinsic difficulty exacerbates the harm of distractors. Several strong models remain ρ -invariant (e.g., gpt-5, o3; $p_{\text{LR}} > 0.2$), and DS-Llama-70B is marginal ($p_{\text{LR}} = 0.055$).
- The $N \times \rho$ interaction is model-dependent and less ubiquitous (significant in 4/22). Two qualitatively distinct regimes appear: for some mid-tier models the coefficient is positive and significant (Qwen3-30B-A3B: $+1.16$, $p_{\text{LR}} = 7.18\text{e-}5$; EXAONE-Deep-32B: $+0.56$, $p_{\text{LR}} = 0.038$), indicating that increasing the needle ratio attenuates length-related degradation (i.e., more signal per distractor helps especially on longer sequences). In contrast, other models show a negative and significant $N \times \rho$ (DS-Qwen-7B: -0.81 , $p_{\text{LR}} = 0.0025$; gemini-2.5-pro: -8.61 , $p_{\text{LR}} = 0.0021$), consistent with long contexts amplifying sensitivity to distractors. Several additional models exhibit borderline positive evidence ($0.05 < p_{\text{LR}} < 0.10$; e.g., Qwen3-32B, Qwen3-8B).

B.2 Three-way effects

A positive three-way interaction $d \times N \times \rho$ is significant in 12/22 models (e.g., DS-Qwen-32B: $+0.36$, $p_{\text{LR}} = 8.1\text{e-}13$; Phi-4-mini-reasoning: $+0.30$, $p_{\text{LR}} = 1.1\text{e-}8$; QwQ-32B: $+0.36$, $p_{\text{LR}} = 3.7\text{e-}12$). Given the quadratic main effect in ρ , this should be read as a moderation of the *slope* with respect to ρ : when tasks are simultaneously long and intrinsically complex, changes in ρ tend to have a larger beneficial marginal effect (i.e., increasing the share of needle statements mitigates the compounded $d \times N$ burden more than when only one of d or N is high). Frontier models show near-zero or slightly negative three-way coefficients (gpt-5: -1.05 , $p_{\text{LR}} = 0.076$; o3: -0.62 , $p_{\text{LR}} = 0.092$), aligning with their weak ρ -dependence overall.

Model	$\beta_{d \times N}$	$p_{LR}(d \times N)$	$\beta_{d \times \rho}$	$p_{LR}(d \times \rho)$	$\beta_{N \times \rho}$	$p_{LR}(N \times \rho)$	$\beta_{d \times N \times \rho}$	$p_{LR}(d \times N \times \rho)$
DeepSeek-R1-0528	-0.03	0.864	0.07	0.89	1.41	0.421	-0.00	0.997
DeepSeek-R1-Distill-Llama-70B	-0.15	$3.41e-05^{***}$	-0.21	0.0552	0.28	0.433	0.14	0.014*
DeepSeek-R1-Distill-Qwen-1.5B	-0.01	0.706	-0.15	0.195	-0.25	0.402	0.08	0.211
DeepSeek-R1-Distill-Qwen-32B	-0.37	$4.07e-32^{***}$	-0.79	$1.84e-17^{***}$	-0.11	0.702	0.36	$8.14e-13^{***}$
DeepSeek-R1-Distill-Qwen-7B	-0.16	$7.63e-07^{***}$	-0.51	$1.41e-07^{***}$	-0.81	0.00246**	0.26	$9.41e-07^{***}$
EXAONE-Deep-32B	-0.27	$1.28e-17^{***}$	-0.89	$2.73e-20^{***}$	0.56	0.0378*	0.42	$2.12e-15^{***}$
Phi-4-mini-reasoning	-0.13	$4.96e-05^{***}$	-0.62	$2.44e-10^{***}$	-0.04	0.886	0.30	$1.08e-08^{***}$
Phi-4-reasoning	-0.45	$1.94e-30^{***}$	-0.50	$2.66e-05^{***}$	0.59	0.0917	0.18	0.00696**
Phi-4-reasoning-plus	-0.41	$4.28e-25^{***}$	-0.39	0.000821**	0.54	0.124	0.15	0.0199*
QwQ-32B	-0.39	$1.25e-33^{***}$	-0.96	$3.77e-23^{***}$	-0.13	0.652	0.36	$3.72e-12^{***}$
Qwen3-1.7B	-0.20	$7.33e-11^{***}$	-0.56	$7.48e-09^{***}$	0.04	0.887	0.27	$3.55e-07^{***}$
Qwen3-30B-A3B	-0.19	$4.57e-09^{***}$	-0.25	0.00919**	1.16	$7.18e-05^{***}$	0.08	0.134
Qwen3-32B	-0.22	$1.76e-10^{***}$	-0.56	$6.88e-08^{***}$	0.53	0.0965	0.21	0.000104**
Qwen3-8B	-0.24	$6.63e-14^{***}$	-0.46	$8.97e-07^{***}$	0.50	0.0767	0.21	$5.36e-05^{***}$
gemini-2.5-flash	-0.58	0.0041**	-0.31	0.681	-3.05	0.133	0.16	0.67
gemini-2.5-flash-lite	-0.27	0.00554**	-0.79	0.00686**	-1.20	0.171	0.31	0.0582
gemini-2.5-pro	-1.84	$7.81e-08^{***}$	-3.53	0.00601**	-8.61	0.00211**	1.61	0.00891**
gpt-5-2025-08-07	-0.13	0.685	1.41	0.276	5.23	0.197	-1.05	0.0762
gpt-5-mini-2025-08-07	-0.56	0.00034**	0.01	0.985	0.48	0.756	-0.06	0.838
gpt-5-nano-2025-08-07	-0.09	0.393	-0.03	0.929	1.82	0.0603	-0.10	0.569
o3-2025-04-16	-0.19	0.353	1.00	0.212	1.72	0.431	-0.62	0.0922
o4-mini-2025-04-16	-0.45	0.00829**	0.12	0.856	-0.05	0.979	-0.07	0.81

Table 1: Interaction analysis across intrinsic difficulty d , effective length $\log_{10} N$, and needle ratio ρ . Coefficients (β) are from a per-model logistic GLM with all pairwise and 3-way interactions plus a quadratic main effect for ρ ; p_{LR} are likelihood-ratio p-values from dropping each interaction from the full model. Stars: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

B.3 Practical implications

Beyond main-effect sensitivities, many models exhibit genuine interaction structure: high d and high N *jointly* depress accuracy more than additively, and high d *amplifies* distractor harm ($d \times \rho < 0$). For a substantial subset of models, increasing the needle share (ρ) yields greater gains specifically when tasks are both long and complex (positive $d \times N \times \rho$), providing a concrete knob for mitigating compounded failure modes. It is important to note however that for models with long-context budget issues (e.g., gemini-2.5-pro; see Appendix F), $N \times \rho$ estimates can be dominated by those failures, inflating effect sizes.

C Model performance with Error-Bars

Due the small size and the large amount of models we compare in Figure 1 of the main paper we omit error-bars to improve legibility. In this section we plot larger versions of the charts in the panel with vertical error bars representing 90% confidence intervals for the mean accuracy of each model-condition pair.

We compute these confidence intervals with the Wilson score method for a binomial proportion. Specifically, for a given point with k correct answers out of n trials we set $\hat{p} = k/n$ and use the 90 % standard-normal quantile $z = 1.644853627$ to obtain

$$[l, u] = \frac{\hat{p} + \frac{z^2}{2n} \pm z \sqrt{\frac{\hat{p}(1-\hat{p}) + z^2/(4n)}{n}}}{1 + \frac{z^2}{n}}.$$

The resulting interval $[l, u]$ marks the range that would contain the true underlying accuracy in 90% of repeated experiments with the same sample size. Wider bars correspond to greater sampling uncertainty, whereas tighter bars

indicate more stable estimates.

The narrow error bars we observe substantiate the discussion of the results in the main paper and highlights a significant difference between the curves at 90% significance and the presence of the characteristic U-shape.

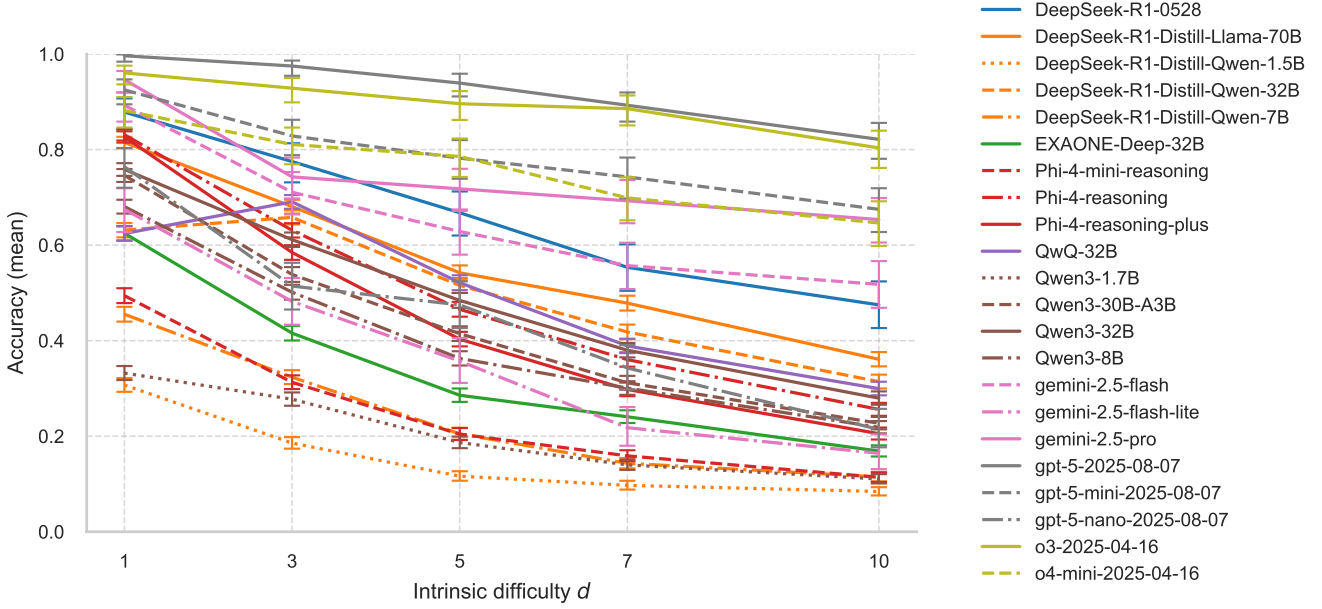


Figure 3: Accuracy relative to difficulty d for $d \in \{1, 3, 5, 7, 10\}$ on the X-Axis

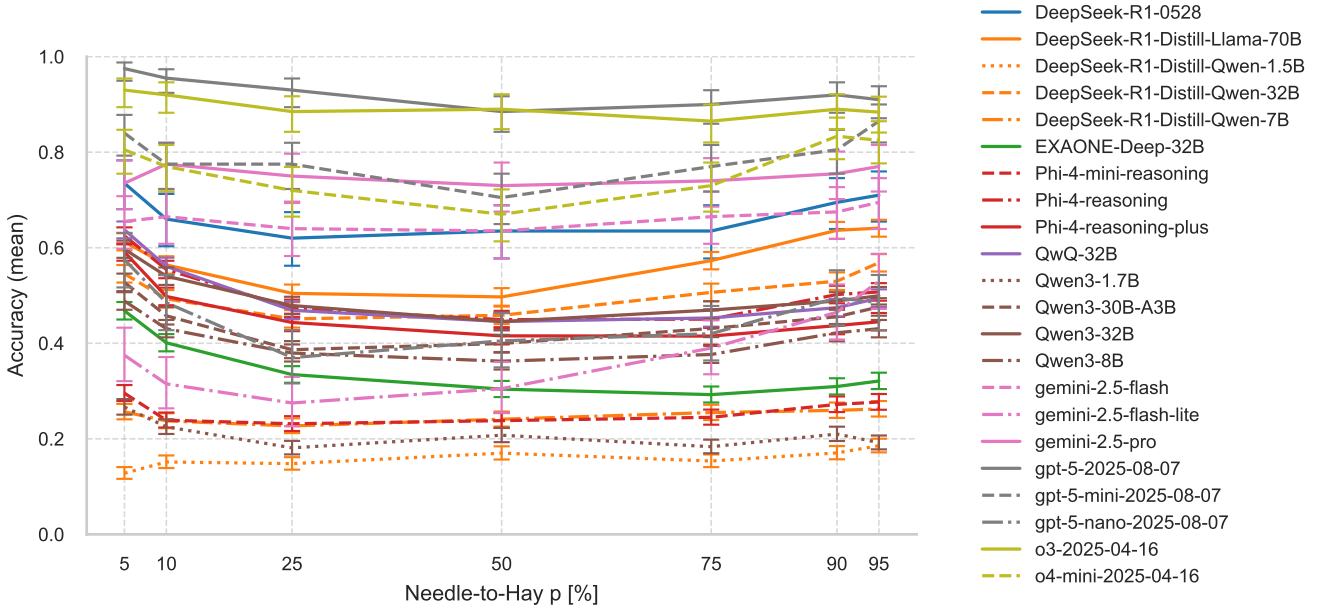


Figure 4: Accuracy relative to distractor-ratio ρ for $\rho \in \{5, 10, 25, 50, 75, 90, 95\}$ on the X-Axis

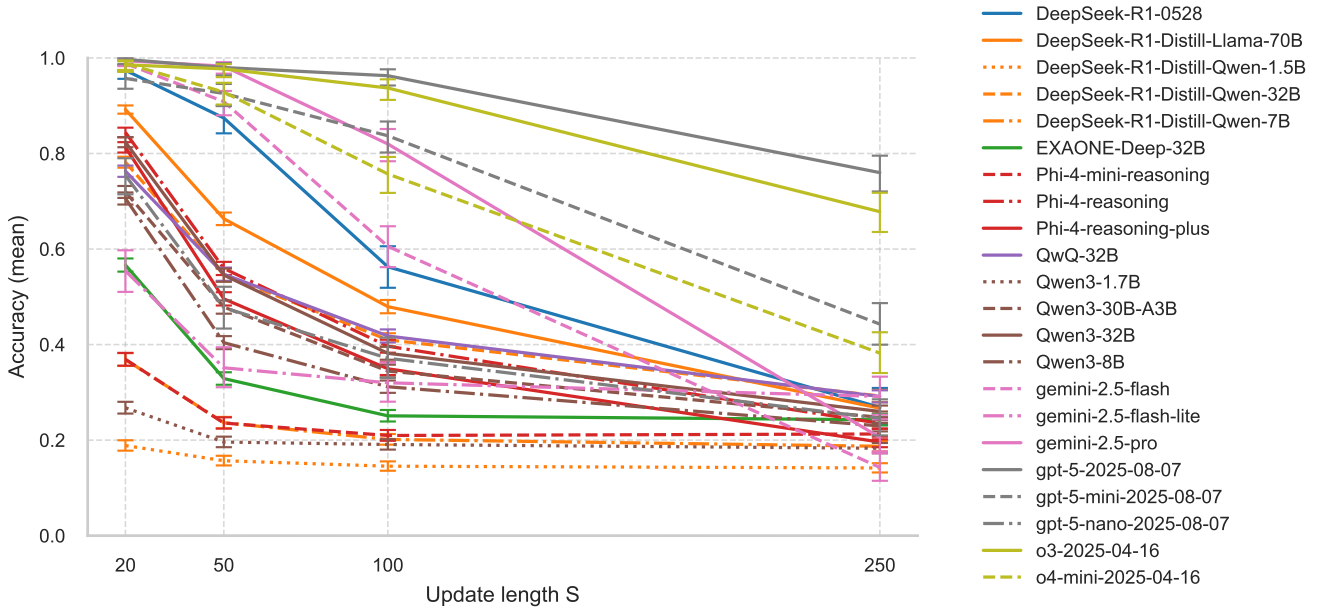


Figure 5: Accuracy relative to total statement length N for $N \in \{20, 50, 100, 250\}$ on the X-Axis

D Evaluation, scoring pipeline, and error taxonomy

This section fully details the evaluation and scoring pipeline functionally in text and in pseudo code.

D.1 Inputs, files, and naming

The scorer consumes three artefacts per puzzle instance z :

1. Model output of the reasoning trace + response (plain text).
2. Puzzle text (for the question): The last line is the natural-language question.
3. Puzzle metadata (JSON state): providing the person-of-interest (PoI), gold label, needles, difficulty, statement count, and category domains.

D.2 Category mapping and qualifiers

The target category is resolved from the last line question using deterministic templates:

- `startswith("Where is")` → `location`
- `startswith("What color shirt")` → `clothes_shirt`
- `startswith("What color pant")` → `clothes_pant`
- `startswith("What color hat")` → `clothes_hat`
- `startswith("What color of socks")` → `clothes_socks`
- `startswith("What color of gloves")` → `clothes_gloves`

- `startswith("What color of underwear") → clothes_underwear`
- `startswith("What is the final hair color") → hair`
- `endswith("most recently eat?") → recent_eat`
- `endswith("recently watch?") → recent_watch`
- `endswith("recently listen to?") → recent_listen`
- `endswith("recently read?") → recent_read`

Qualifiers (used to find a “valid PoI sentence”):

- `location: {at, located, in}`
- `clothes_shirt: {shirt, wear}; clothes_pant: {pant, wear}; clothes_hat: {hat, wear}; clothes_gloves: {glove, wear}; clothes_socks: {sock, wear}; clothes_underwear: {underwear, wear}`
- `hair: {hair}`
- `recent_eat: {eat, ate}; recent_watch: {watch, watched, movie}; recent_listen: {listen, listened, music}; recent_read: {read, book}`

The PoI name and the domain values for the target category are read from the puzzle metadata file.

D.3 Normalization, sentence windows, and context budget

Normalization:

- Lowercase the full model output; split by newlines.
- If the last line starts with “(” and ends with “)”, drop that line because it’s usually just a clarification that follows the final answer.
- Drop empty lines; if no lines remain, keep a single empty line.

Sentence windows:

1. **valid_poi_last_sentence**: last line containing PoI and any category qualifier; take `TakeLastSentence`.
2. **poi_last_sentence**: last line containing PoI; take `TakeLastSentence` of that line.
3. **last_sentence**: apply `TakeLastSentence` to the last remaining line; see below.

`TakeLastSentence(s)`: let $p = \text{split}(s, ".")$. If $\text{len}(p) \geq 2$, return $p[-2]$; else return s .

Context limit failure: we mark `wrong_max_context` if any of the following holds:

$$\text{prompt_tokens} + \text{response_tokens} + 20 \geq 32'768, \quad \text{or} \quad \text{last_sentence} = "", \quad \text{or} \quad \text{empty output.}$$

The 20 additional tokens added to the `response_tokens` is necessary because when models usually produce incomplete responses a few tokens before the maximum-output-token limit they were prompted to decode with.

D.4 Matching, synonyms, and alternatives

Since some models respond with the answer attribute starting with a special character instead of an empty space (like a word) the containment accepts the following alternative word boundaries (code `CheckAnswer(term, sentence)` returns true if any holds):

- `sentence.startswith(term)` or `sentence == term`
- substring appears prefixed by one of: space, “[”, “””, “*”, “_”, “{”, “(”

Since some attributes in the attribute taxonomy could be spelled alternatively, we relax Synonym whitelist for them in the evaluation (code `CheckCorrectAnswer(gold, sentence)`):

- `reggae`: accept `reggae` or `reggae`
- `sci-fi`: accept `sci-fi`, `science fiction`, `science-fiction`
- `camp`: accept `camp`, `campground`
- `potatoes`: accept `potatoes`, `potato`
- `market`: accept `market`, `marketplace`
- `livingroom`: accept `livingroom`, `living room`
- otherwise: accept exactly `gold` by boundary-tolerant containment

We further ensure no alternative value is present in the last sentence and therefore compute (`AltFlag(sentence, alts, gold)`) as:

1. Let $s = \text{lowercase}(\text{sentence})$, $g = \text{lowercase}(\text{gold})$, and $A = \text{lowercase alternatives} = \text{domain values} \setminus \{g\}$.
2. Define `present(t) = CheckCorrectAnswer(t, s)`.
3. If no alternative is present, return false.
4. If both `gold` and at least one alternative are present, keep the alternative flag only if some alternative’s character span encloses the `gold`’s span in s (“spans” test); otherwise clear the flag. This is done to accommodate attribute values like ‘non-fiction’ which contain alternative values like ‘fiction’.
5. Since sometimes models respond by mentioning also previous value of the attribute (eg a sentence like “... the final step explicitly converts gray to green, resulting in **green**”) we do not set the flag if the `gold` flag is the last mentioned attribute in the sentence.

D.5 Buckets, precedence, and accuracy

Let:

```
c_valid = CheckCorrectAnswer(gold, valid_poi_last_sentence),
c_poi = CheckCorrectAnswer(gold, poi_last_sentence),
c_last = CheckCorrectAnswer(gold, last_sentence),
a_valid = AltFlag(valid_poi_last_sentence, alts, gold),
a_poi = AltFlag(poi_last_sentence, alts, gold),
a_last = AltFlag(last_sentence, alts, gold).
```

Classification (first applicable wins):

1. if c_{valid} and not a_{valid} : `correct_valid`
2. else if c_{poi} and not a_{poi} : `correct_poi`
3. else if c_{last} and not a_{last} : `correct_last_sentence`
4. else if not c_{valid} and a_{valid} and `valid_poi_last_sentence` \neq `""`: `wrong_logic`
5. else if not c_{poi} and a_{poi} and `poi_last_sentence` \neq `""`: `wrong_logic_poi`
6. else if not c_{last} and a_{last} and `last_sentence` \neq `""`: `wrong_logic_last_sentence`
7. else: `wrong_other`

Note: When a sentence contains both gold and an alternative that spans the gold, this does not trigger a `wrong_logic*` bucket unless the corresponding “not c_{\cdot} and a_{\cdot} ” guard holds; such cases can fall through to `wrong_other`, matching the implementation. The main figures/regressions of the paper use accuracy with `correct` \in `{correct_valid, correct_poi, correct_last_sentence}` and `correct` as the complement.

D.6 Pseudocode

Using pseudo code, figure 1 explains the data preparation, figure 2 explains the scoring of the results and success/failure types, and figure 3 details the used utility functions.

Algorithm 1: Pseudocode of the data preparation.

```

// Parse identifiers and gold from puzzle name: N.d.needles.puzzleId.gold
(N, d, needles, puzzleId, gold)  $\leftarrow$  PARSE(ans_path);
// Load question and resolve category
question  $\leftarrow$  LAST_LINE(puzzle_txt_path);
category  $\leftarrow$  MapQuestionToCategory(question);
// Load metadata needed for evaluation
(poi, domains)  $\leftarrow$  READJSON(state_json_path);
poi  $\leftarrow$  LOWER(poi); domainValues  $\leftarrow$  domains[category];
quals  $\leftarrow$  GetQualifiersFor(category);
// Read and normalize model output
resp  $\leftarrow$  READ(ans_path);
lines  $\leftarrow$  SPLIT_LINES(LOWER(resp));
if STARTSWITH(lines[-1], "(")  $\wedge$  ENDSWITH(lines[-1], ")") then
   $\lfloor$  DROP_LAST(lines)

lines  $\leftarrow$  FILTER_NONEMPTY(lines);
if LEN(lines) = 0 then
   $\lfloor$  lines  $\leftarrow$  [""]
```

E Error distribution

The following tables illustrate the error-type distribution per difficulty 2, and length 3. The purpose of this distinction is to illustrate three primary error types (i.e., long context error, logic error, other error) and to distinguish among the logic errors how well the model followed the instruction in its response formatting. The other error captures cases where the model responds but doesn’t mention any attribute, usually this is cases where the model erroneously

assumes the problem can't be solved with the provided information. For conciseness the error types in the table headers are abbreviated as following:

- Wrong max-context \rightarrow ctx
- Wrong logic (last valid POI sentence) \rightarrow valid-logic
- Wrong logic (last POI sentence) \rightarrow poi-logic
- Wrong logic (last sentence) \rightarrow last-logic
- Wrong other \rightarrow other

If the total number of evaluated samples does not precisely match the expected number of samples (e.g 'o3-2025-04-16' with difficulty 5 resulted in 279 vs 280 expected examples), this usually indicates the puzzle was not processed due to a an API error (e.g. a security refusal). These few missing cases were attempted to be evaluated a second time but refused again.

Table 2: Error-type distribution by difficulty d per model (aggregated over N and ρ).

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
DeepSeek-R1-0528	1	280	88	3	31	0	0	0
DeepSeek-R1-0528	3	280	78	11	52	0	0	0
DeepSeek-R1-0528	5	280	67	13	80	0	0	0
DeepSeek-R1-0528	7	280	55	16	109	0	0	0
DeepSeek-R1-0528	10	280	48	19	128	0	0	0
DeepSeek-R1-Distill-Llama-70B	1	2800	82	0	507	0	2	9
DeepSeek-R1-Distill-Llama-70B	3	2800	68	11	877	0	3	3
DeepSeek-R1-Distill-Llama-70B	5	2800	54	12	1260	0	8	2
DeepSeek-R1-Distill-Llama-70B	7	2800	48	28	1424	0	4	4
DeepSeek-R1-Distill-Llama-70B	10	2800	36	33	1747	0	4	5
DeepSeek-R1-Distill-Qwen-1.5B	1	2800	31	561	1140	4	81	154
DeepSeek-R1-Distill-Qwen-1.5B	3	2800	19	543	1305	3	82	347
DeepSeek-R1-Distill-Qwen-1.5B	5	2800	12	613	1381	4	63	413
DeepSeek-R1-Distill-Qwen-1.5B	7	2800	10	695	1323	7	60	443
DeepSeek-R1-Distill-Qwen-1.5B	10	2800	8	717	1371	7	62	407
DeepSeek-R1-Distill-Qwen-32B	1	2800	63	1	1023	0	3	5
DeepSeek-R1-Distill-Qwen-32B	3	2800	66	46	908	0	2	1
DeepSeek-R1-Distill-Qwen-32B	5	2800	52	78	1275	0	1	2
DeepSeek-R1-Distill-Qwen-32B	7	2800	42	94	1532	0	0	3
DeepSeek-R1-Distill-Qwen-32B	10	2800	31	106	1813	0	0	1
DeepSeek-R1-Distill-Qwen-7B	1	2800	46	47	1454	5	4	15
DeepSeek-R1-Distill-Qwen-7B	3	2800	32	124	1561	5	70	134
DeepSeek-R1-Distill-Qwen-7B	5	2800	20	246	1706	3	65	206
DeepSeek-R1-Distill-Qwen-7B	7	2800	14	317	1802	8	49	223
DeepSeek-R1-Distill-Qwen-7B	10	2800	12	352	1770	6	60	290
EXAONE-Deep-32B	1	2800	62	0	995	5	2	48
EXAONE-Deep-32B	3	2800	42	2	1595	2	27	10
EXAONE-Deep-32B	5	2800	29	10	1938	2	45	5
EXAONE-Deep-32B	7	2800	24	14	2060	1	42	9
EXAONE-Deep-32B	10	2800	17	16	2257	1	40	13

Table 2: Error-type distribution by difficulty d per model (aggregated over N and ρ). (continued)

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
gemini-2.5-flash	1	280	89	6	24	0	0	0
gemini-2.5-flash	3	280	71	64	17	0	0	0
gemini-2.5-flash	5	280	63	93	11	0	0	0
gemini-2.5-flash	7	280	56	116	8	0	0	0
gemini-2.5-flash	10	280	52	110	25	0	0	0
gemini-2.5-flash-lite	1	280	68	0	91	0	0	0
gemini-2.5-flash-lite	3	280	48	5	138	0	0	2
gemini-2.5-flash-lite	5	280	36	3	176	0	0	1
gemini-2.5-flash-lite	7	280	22	8	209	0	0	2
gemini-2.5-flash-lite	10	280	16	1	227	2	0	4
gemini-2.5-pro	1	280	95	10	5	0	0	0
gemini-2.5-pro	3	280	74	69	3	0	0	0
gemini-2.5-pro	5	280	72	71	8	0	0	0
gemini-2.5-pro	7	280	69	79	7	0	0	0
gemini-2.5-pro	10	280	65	88	9	0	0	0
gpt-5-2025-08-07	1	280	100	0	1	0	0	0
gpt-5-2025-08-07	3	280	98	0	7	0	0	0
gpt-5-2025-08-07	5	280	94	0	17	0	0	0
gpt-5-2025-08-07	7	280	89	3	26	0	0	1
gpt-5-2025-08-07	10	280	82	29	21	0	0	0
gpt-5-mini-2025-08-07	1	280	92	0	14	0	0	7
gpt-5-mini-2025-08-07	3	280	83	0	46	0	0	2
gpt-5-mini-2025-08-07	5	280	78	3	52	0	1	5
gpt-5-mini-2025-08-07	7	280	74	12	58	0	0	2
gpt-5-mini-2025-08-07	10	280	68	27	63	0	0	1
gpt-5-nano-2025-08-07	1	280	76	0	65	1	0	0
gpt-5-nano-2025-08-07	3	280	51	0	133	0	3	0
gpt-5-nano-2025-08-07	5	280	48	0	145	0	2	0
gpt-5-nano-2025-08-07	7	280	34	0	183	0	1	0
gpt-5-nano-2025-08-07	10	280	21	0	219	0	1	0
o3-2025-04-16	1	279	96	0	11	0	0	0
o3-2025-04-16	3	280	93	0	20	0	0	0
o3-2025-04-16	5	279	90	1	28	0	0	0
o3-2025-04-16	7	280	89	2	30	0	0	0
o3-2025-04-16	10	280	80	21	34	0	0	0
o4-mini-2025-04-16	1	279	88	0	31	0	0	2
o4-mini-2025-04-16	3	280	81	0	53	0	0	0
o4-mini-2025-04-16	5	280	79	3	57	0	0	0
o4-mini-2025-04-16	7	279	70	5	78	0	0	1
o4-mini-2025-04-16	10	280	65	13	86	0	0	0
Phi-4-mini-reasoning	1	2800	49	4	1222	16	48	126
Phi-4-mini-reasoning	3	2800	31	1	1613	6	126	178
Phi-4-mini-reasoning	5	2800	20	12	1733	8	286	189
Phi-4-mini-reasoning	7	2800	16	7	1785	6	354	203
Phi-4-mini-reasoning	10	2800	11	18	1856	5	415	187
Phi-4-reasoning	1	2800	83	14	417	0	0	42

Table 2: Error-type distribution by difficulty d per model (aggregated over N and ρ). (continued)

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
Phi-4-reasoning	3	2800	63	79	921	0	0	33
Phi-4-reasoning	5	2800	47	246	1219	1	0	30
Phi-4-reasoning	7	2800	36	522	1240	0	0	29
Phi-4-reasoning	10	2800	26	857	1200	1	0	25
Phi-4-reasoning-plus	1	2800	83	100	347	1	0	37
Phi-4-reasoning-plus	3	2800	58	182	938	0	0	43
Phi-4-reasoning-plus	5	2800	40	471	1153	1	0	46
Phi-4-reasoning-plus	7	2800	30	936	1001	2	0	27
Phi-4-reasoning-plus	10	2800	21	1198	994	0	1	32
Qwen3-1.7B	1	2800	33	1	1840	1	11	17
Qwen3-1.7B	3	2800	28	0	2010	0	2	11
Qwen3-1.7B	5	2800	19	0	2268	0	1	8
Qwen3-1.7B	7	2800	14	0	2396	0	2	10
Qwen3-1.7B	10	2798	11	0	2483	0	1	5
Qwen3-30B-A3B	1	2800	75	18	652	3	0	38
Qwen3-30B-A3B	3	2800	54	48	1237	1	0	6
Qwen3-30B-A3B	5	2800	41	40	1587	2	0	10
Qwen3-30B-A3B	7	2800	31	25	1894	2	0	6
Qwen3-30B-A3B	10	2800	23	19	2133	3	1	8
Qwen3-32B	1	2800	76	0	626	0	1	49
Qwen3-32B	3	2800	61	1	1075	1	1	10
Qwen3-32B	5	2800	48	0	1434	1	3	6
Qwen3-32B	7	2800	38	0	1721	0	6	9
Qwen3-32B	10	2800	28	1	1994	0	12	10
Qwen3-8B	1	2800	68	77	739	2	0	76
Qwen3-8B	3	2800	50	136	1190	3	18	49
Qwen3-8B	5	2800	36	106	1619	3	19	37
Qwen3-8B	7	2800	30	101	1791	0	27	39
Qwen3-8B	10	2800	22	78	2046	0	39	25
QwQ-32B	1	2800	62	2	1044	1	0	5
QwQ-32B	3	2800	69	12	850	0	3	1
QwQ-32B	5	2800	52	52	1279	0	3	6
QwQ-32B	7	2800	39	112	1596	0	1	2
QwQ-32B	10	2800	30	273	1674	0	5	9

Table 3: Error-type distribution by puzzle length N per model (aggregated over d and ρ).

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
DeepSeek-R1-0528	20	350	97	7	2	0	0	0
DeepSeek-R1-0528	50	350	87	14	30	0	0	0
DeepSeek-R1-0528	100	350	56	25	128	0	0	0
DeepSeek-R1-0528	250	350	27	16	240	0	0	0
DeepSeek-R1-Distill-Llama-70B	20	3500	89	0	372	0	1	4

Table 3: Error-type distribution by puzzle length N per model (aggregated over d and ρ). (continued)

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
DeepSeek-R1-Distill-Llama-70B	50	3500	66	2	1167	0	4	5
DeepSeek-R1-Distill-Llama-70B	100	3500	48	2	1811	0	6	3
DeepSeek-R1-Distill-Llama-70B	250	3500	27	80	2465	0	10	11
DeepSeek-R1-Distill-Qwen-1.5B	20	3500	19	749	1676	8	58	349
DeepSeek-R1-Distill-Qwen-1.5B	50	3500	16	752	1749	5	85	360
DeepSeek-R1-Distill-Qwen-1.5B	100	3500	15	698	1749	4	90	450
DeepSeek-R1-Distill-Qwen-1.5B	250	3500	14	930	1346	8	115	605
DeepSeek-R1-Distill-Qwen-32B	20	3500	78	1	760	0	0	2
DeepSeek-R1-Distill-Qwen-32B	50	3500	55	1	1583	0	2	3
DeepSeek-R1-Distill-Qwen-32B	100	3500	41	0	2061	0	1	3
DeepSeek-R1-Distill-Qwen-32B	250	3500	29	323	2147	0	3	4
DeepSeek-R1-Distill-Qwen-7B	20	3500	37	117	2012	4	14	63
DeepSeek-R1-Distill-Qwen-7B	50	3500	24	207	2298	1	43	123
DeepSeek-R1-Distill-Qwen-7B	100	3500	20	311	2177	11	75	221
DeepSeek-R1-Distill-Qwen-7B	250	3500	19	451	1806	11	116	461
EXAONE-Deep-32B	20	3500	57	7	1451	4	35	20
EXAONE-Deep-32B	50	3500	33	17	2255	2	56	19
EXAONE-Deep-32B	100	3500	25	7	2556	4	34	21
EXAONE-Deep-32B	250	3500	24	11	2583	1	31	25
gemini-2.5-flash	20	350	99	0	4	0	0	0
gemini-2.5-flash	50	350	91	10	22	0	0	0
gemini-2.5-flash	100	350	61	99	39	0	0	0
gemini-2.5-flash	250	350	14	280	20	0	0	0
gemini-2.5-flash-lite	20	350	55	0	154	0	0	2
gemini-2.5-flash-lite	50	350	35	0	223	1	0	3
gemini-2.5-flash-lite	100	350	32	1	235	0	0	2
gemini-2.5-flash-lite	250	350	29	16	229	1	0	2
gemini-2.5-pro	20	350	99	0	2	0	0	0
gemini-2.5-pro	50	350	98	1	5	0	0	0
gemini-2.5-pro	100	350	82	48	15	0	0	0
gemini-2.5-pro	250	350	21	268	10	0	0	0
gpt-5-2025-08-07	20	350	100	0	1	0	0	0
gpt-5-2025-08-07	50	350	98	0	7	0	0	0
gpt-5-2025-08-07	100	350	96	0	12	0	0	1
gpt-5-2025-08-07	250	350	76	32	52	0	0	0
gpt-5-mini-2025-08-07	20	350	96	0	12	0	0	3
gpt-5-mini-2025-08-07	50	350	93	0	23	0	0	3
gpt-5-mini-2025-08-07	100	350	84	0	54	0	0	3
gpt-5-mini-2025-08-07	250	350	44	42	144	0	1	8
gpt-5-nano-2025-08-07	20	350	75	0	86	0	0	0
gpt-5-nano-2025-08-07	50	350	48	0	181	1	1	0
gpt-5-nano-2025-08-07	100	350	37	0	217	0	3	0
gpt-5-nano-2025-08-07	250	350	25	0	261	0	3	0
o3-2025-04-16	20	350	99	0	5	0	0	0
o3-2025-04-16	50	350	98	0	8	0	0	0
o3-2025-04-16	100	350	94	0	22	0	0	0

Table 3: Error-type distribution by puzzle length N per model (aggregated over d and ρ). (continued)

Model	Dim	Samples	Acc [%]	ctx	valid-logic	poi-logic	last-logic	other
o3-2025-04-16	250	348	68	24	88	0	0	0
o4-mini-2025-04-16	20	350	99	0	3	0	0	1
o4-mini-2025-04-16	50	350	93	0	24	0	0	1
o4-mini-2025-04-16	100	350	76	0	85	0	0	0
o4-mini-2025-04-16	250	348	38	21	193	0	0	1
Phi-4-mini-reasoning	20	3500	37	8	1765	7	213	215
Phi-4-mini-reasoning	50	3500	24	4	2163	11	289	208
Phi-4-mini-reasoning	100	3500	21	13	2198	6	327	222
Phi-4-mini-reasoning	250	3500	21	17	2083	17	400	238
Phi-4-reasoning	20	3500	84	11	496	0	0	38
Phi-4-reasoning	50	3500	56	106	1393	0	0	43
Phi-4-reasoning	100	3500	40	332	1736	1	0	43
Phi-4-reasoning	250	3500	24	1269	1372	1	0	35
Phi-4-reasoning-plus	20	3500	81	12	586	1	0	55
Phi-4-reasoning-plus	50	3500	50	382	1326	3	0	54
Phi-4-reasoning-plus	100	3500	35	668	1563	0	1	45
Phi-4-reasoning-plus	250	3500	20	1825	958	0	0	31
Qwen3-1.7B	20	3500	27	0	2546	0	5	12
Qwen3-1.7B	50	3500	20	1	2796	1	3	13
Qwen3-1.7B	100	3500	19	0	2818	0	6	8
Qwen3-1.7B	250	3498	18	0	2837	0	3	18
Qwen3-30B-A3B	20	3500	72	25	942	1	1	12
Qwen3-30B-A3B	50	3500	48	40	1764	2	0	19
Qwen3-30B-A3B	100	3500	34	44	2228	1	0	21
Qwen3-30B-A3B	250	3500	25	41	2569	7	0	16
Qwen3-32B	20	3500	82	0	607	0	1	8
Qwen3-32B	50	3500	55	1	1577	0	3	10
Qwen3-32B	100	3500	38	0	2125	0	4	35
Qwen3-32B	250	3500	26	1	2541	2	15	31
Qwen3-8B	20	3500	71	75	912	0	18	24
Qwen3-8B	50	3500	40	108	1893	1	31	53
Qwen3-8B	100	3500	31	117	2198	3	27	64
Qwen3-8B	250	3500	23	198	2382	4	27	85
QwQ-32B	20	3500	76	8	810	0	4	7
QwQ-32B	50	3500	55	27	1550	0	1	6
QwQ-32B	100	3500	42	36	1991	0	5	5
QwQ-32B	250	3500	29	380	2092	1	2	5

F AIC-Comparison

In this section we compare the Akaike Information Criterion (AIC) of the GLM model containing a pure linear ρ term with a GLM model with a squared term added for ρ . The equations for the models are as following with $Y = 1$ indicating a correctly solved puzzle:

Linear model

$$\Pr(Y=1) = \sigma(\beta_0 + \beta_d d + \beta_N \log_{10} N + \beta_\rho \rho),$$

Quadratic model

$$\Pr(Y=1) = \sigma(\beta_0 + \beta_d d + \beta_N \log_{10} N + \beta_\rho \rho + \beta_{\rho^2} \rho^2),$$

To assess whether the quadratic specification provides a statistically significant improvement over the linear GLM, we compare their maximised log-likelihoods ℓ_{lin} and ℓ_{quad} . The likelihood-ratio statistic $D = 2(\ell_{\text{quad}} - \ell_{\text{lin}})$ follows, under the null hypothesis that the extra term is unnecessary, a chi-squared distribution with one degree of freedom because the quadratic model introduces exactly one additional parameter.

The p -value reported in Table 4 is the upper-tail probability $p = \Pr(\chi_1^2 \geq D)$. p -values below 0.05 indicate that the quadratic term yields a statistically significant gain in fit and therefore justifies its inclusion.

We find that for all except for two models (i.e., DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B), the quadratic term for ρ results in a significantly improved AIC value. Therefore we include the quadratic ρ term in the GLM specification within the paper.

G Complete Attribute Ontology

The following section lists the full attribute ontology of all values available for the categories.

people Peter, Paul, Mary, John, Mark, Jeff, Craig, Daniel, Anna, Arnoldo, Ali, Benjamin, Joe, Donald, Mitch, Chuck, Jack, Lucas, Jeniffer, Adam, Greg, Allan, David, Ellen, Fred, Hank, Hubert, Ian, Ingrid, Rebecca, Ken, Lewis, Michael, Nathaniel, Oliver, Russ, Steve, Sandy, Ted, Tanya, Veronica, Vincent, Wesley, Brad, Sam, Igor, Sue, Jan, Jeffrey, Jacques, Debby, Olivia, Benedict, Chris, Charles, Harry, Eli, Mahmoud, Chen, William, Linda, Elizabeth, Robert, Jennifer, Emily, Joseph, Thomas, Patricia, Anthony, Jessica, Brian, Lisa, Kevin, Karen, Laura, Eric, Stephanie, Michelle, George, Andrew, Joshua, Amber, Timothy, Victoria, Richard, Cynthia, Brandon, Megan, Matthew, Nancy, Jacqueline, Gary, Dorothy, Edward, Kimberly, Scott, Sara, Justin, Brittany, Ronald, Deborah, Janet, Christopher, Alexander, Samantha, Oscar, Cindy, Frank, Carl, Paula, Irene, Theresa, Dennis, Ralph, Gerald, Martin, Terry, Bryan, Lance, Corey, Casey, Brent, Derek, Travis, Austin, Victor, Jesse, Zachary, Kyle, Aaron, Betty, Connie, Holly, Donna, Gloria, Carla, Isabel, Sylvia, Evelyn, Doris, Arthur, Raymond, Harold, Lawrence, Neil, Brenda, Tracy, Simon, Wendy, Zoe, Ethan, Calvin, Sean, Ruth, Sheila, Miriam, Lorraine, Fay, Sophie

clothes_socks blue, red, yellow, green, purple, pink, orange, black, white, gray

clothes_shirt blue, red, yellow, green, purple, pink, orange, black, white, gray

clothes_pant blue, red, yellow, green, purple, pink, orange, black, white, gray

clothes_hat blue, red, yellow, green, purple, pink, orange, black, white, gray

clothes_gloves blue, red, yellow, green, purple, pink, orange, black, white, gray

clothes_underwear blue, red, yellow, green, purple, pink, orange, black, white, gray

hair blue, red, yellow, green, purple, pink, orange, black, white, gray

recent_eat pizza pasta burrito sushi taco burger toast egg banana potatoes

recent_listen rock, pop, country, electronic, folk, jazz, blues, classical, funk, ska, rap, synth, disco, reaggea

recent_watch drama, comedy, thriller, romance, adventure, horror, sci-fi, action, western, fantasy, documentary, mystery, crime, musical

recent_read fiction, mystery, novel, thriller, biography, sci-fi, non-fiction, essay, encyclopedia, dictionary

location bathroom livingroom kitchen basement toilet balcony garden pool bedroom store university farm office bank tree museum school airport zoo train bus park butcher library restaurant mall mountain tunnel church river pond harbor taxi gallery bar pizzeria beach gym elevator insurance embassy police hospital festival monument laboratory observatory valley motorway viewpoint synagogue factory castle cave stadium arena cabin plaza amphitheater bridge pier vineyard forest cliff desert creek bay lighthouse orchard resort camp inn motel aquarium bazaar chapel monastery lookout campground retreat dock depot consulate manor theatre cathedral casino lodge mill bakery spa station diner gazebo terrace arcade boardwalk winery hill plateau ridge port oasis market fairground quarry mine grove auditorium cemetery dunes courthouse prison fort granary ranch promenade coliseum field tower pavilion silo bistro labyrinth cafe saloon brewery carnival marina estate safari cottage courtyard waterpark island greenhouse meadow lagoon ford hacienda village marketplace grotto maze golfcourse atrium academy waterfront peninsula cove summit plains

H Token-length distributions by task length (N) and difficulty (d)

We report token-length distributions per model tokenizer across the full CogniLoad grid in Table 5, Table 6, Table 7, and Table 8. For each model and each (N, d) condition, we show the mean and the 90th percentile (p90) of both the input (prompt) tokens and the output (reasoning) tokens, compactly formatted as Pmean[P_{p90}] — Omean[O_{p90}]. For example, a cell like P3995[4683] — O14778[25212] indicates that the mean prompt length is ≈ 3995 tokens with $p90 \approx 4683$, and the mean output length is about ≈ 14778 tokens with $p90 \approx 25212$ for that condition. Presenting both prompt and output side by side makes it possible to separate the load imposed by the puzzle itself from model and tokenizer-specific verbosity in decoding.

Across all models, prompt length grows monotonically with N and increases with d, as expected from the construction of the puzzles. At fixed (N, d), the variation in prompt tokens across tokenizers is small (i.e. typically a few percent) which indicates that models received essentially the same input burden for a given condition. For example, at N = 250 and d = 10 as the highest load condition, prompt means cluster around 16000–17000 tokens, while at d = 1 they are around 3500–4000 with these patterns recurring for every tokenizer. This confirms that differences in performance at fixed (N, d) do not stem from large differences in presented input length.

On the contrary, output lengths vary substantially across models and show heavy-tailed behavior in several cases, especially at larger N. Gemini 2.5 models tend to produce the longest outputs: at N = 250 and moderate d, their output p90 values are frequently in the mid-20000s while prompt p90 values are often around 6500–7000, so totals approach or exceed the 32768-token budget. For instance, at N = 250 and d = 5, gemini-2.5-pro shows approximately P9474[10084] — O23221[24321], and gemini-2.5-flash is in a similar range. These totals explain the higher rates of max-context overflows we observed for Gemini at high N indicating a serving limitation rather than genuine reasoning error. By comparison, o3 and the gpt-5 family are more restrained in their output lengths at the same (N, d): their output p90s are typically around 11000–19000 at N = 250 depending on d, leaving more headroom when paired with prompt p90s near 6500–7000. Mid-tier open models such as DeepSeek-R1-Distill-Llama-70B, Qwen3-32B, QwQ-32B, and EXAONE-Deep-32B generally produce shorter outputs still, for example, Qwen3-32B at N = 250 and d = 7 shows roughly P12419[13055] — O5034[8153], a total far below the context limit. Smaller models often have shorter outputs on average but can occasionally exhibit very long tails. Inspecting their transparent reasoning traces we notice those outliers being consistent with the error taxonomy’s degeneracy categories (e.g., wrong-max-context or ”other”), where models infinitely repeat tokens or drift into infinite reasoning loops rather than concluding. One peculiar case is Gemini-2.5-flash-lite at the cell N=250/d=10 resulting in P16683[17372] — O366[9]. Here this particular model doesn’t even try to reason through the problem but usually just guesses the response with barely any reasoning.

Taken together, these distributions show that prompt length scales predictably with N and d and is closely matched across tokenizers, while output length is where models differ. In our runs, most models and conditions—including

many at $N = 250$ —remain well below 32768 tokens in total, indicating that CogniLoad puzzles are generally solvable within a 32K context. Overflows are concentrated in specific settings where certain models (notably Gemini 2.5 at high N) produce heavy-tailed outputs while smaller models show occasional outliers that, when inspecting the reasoning traces, boil down to infinite reasoning loops and incoherence issues that repeat the same tokens indefinitely. Consequently, for the majority of models and conditions the observed performance trends at long N reflect reasoning under load rather than system ceilings.

Algorithm 2: Pseudocode of the CogniLoad scoring pipeline.

```
// Prepare data
// Extract windows
last_sentence ← TakeLastSentence(lines[-1]);
poi_lines ← {s ∈ lines : CONTAINS(s, poi)};
poi_last ← LASTOREMPTY(poi_lines);
valid_poi_lines ← {s ∈ poi_lines : CONTAINSANY(s, quals)};
valid_poi_last ← LASTOREMPTY(valid_poi_lines);
// Budget/emptiness guard
if ExceededContext(runtime_meta) ∨ last_sentence = "" ∨ LEN(resp) = 0 then
  ⊥ return wrong_max_context
// Gold matches
c_valid ← CheckCorrectAnswer(gold, valid_poi_last);
c_poi ← CheckCorrectAnswer(gold, poi_last);
c_last ← CheckCorrectAnswer(gold, last_sentence);
// Alternative flags with span rule
alts ← {v ∈ domainValues : v ≠ gold};
a_valid ← AltFlag(valid_poi_last, alts, gold);
a_poi ← AltFlag(poi_last, alts, gold);
a_last ← AltFlag(last_sentence, alts, gold);
// Buckets (first match wins)
if c_valid ∧ ¬a_valid then
  ⊥ return correct_valid
else if c_poi ∧ ¬a_poi then
  ⊥ return correct_poi
else if c_last ∧ ¬a_last then
  ⊥ return correct_last_sentence
else if ¬c_valid ∧ a_valid ∧ valid_poi_last ≠ "" then
  ⊥ return wrong_logic
else if ¬c_poi ∧ a_poi ∧ poi_last ≠ "" then
  ⊥ return wrong_logic_poi
else if ¬c_last ∧ a_last ∧ last_sentence ≠ "" then
  ⊥ return wrong_logic_last_sentence
else
  ⊥ return wrong_other
```

Algorithm 3: Pseudocode of the CogniLoad scoring pipeline utility functions.

Utilities.

ExceededContext: return true if `prompt_tokens + response_tokens + 20 ≥ 32 768`.

TakeLastSentence(s): let $p = \text{SPLIT}(s, ".")$. If $\text{LEN}(p) \geq 2$, return $p[-2]$; else s .

CheckCorrectAnswer($gold, s$):

- **CheckAnswer(t, s)** is true if any holds: $\text{STARTSWITH}(s, t)$, $s = t$, or s contains one of: "space+t", "[t", ""t", "*t", "_t", "{ t", "(t".
- If $gold = \text{reaggea}$: return **CheckAnswer**(reaggea) \vee **CheckAnswer**(reggae).
- If $sci\text{-}fi$: return **CheckAnswer**(sci-fi) \vee **CheckAnswer**(science fiction) \vee **CheckAnswer**(science-fiction).
- If $camp$: **CheckAnswer**(camp) \vee **CheckAnswer**(campground).
- If $potatoes$: **CheckAnswer**(potatoes) \vee **CheckAnswer**(potato).
- If $market$: **CheckAnswer**(market) \vee **CheckAnswer**(marketplace).
- If $livingroom$: **CheckAnswer**(livingroom) \vee **CheckAnswer**(living room).
- Else: **CheckAnswer**(gold).

AltFlag($s, alts, gold$):

1. $s \leftarrow \text{LOWER}(s)$, $g \leftarrow \text{LOWER}(gold)$, $A \leftarrow \{\text{LOWER}(a) : a \in alts, a \neq g\}$.
 2. $\text{alt_present} \leftarrow \bigvee_{a \in A} \text{CheckCorrectAnswer}(a, s)$.
 3. $g_idx \leftarrow s.\text{RFIND}(g)$.
 4. **if** $g_idx \neq -1$ and alt_present : let $g_end = g_idx + \text{LEN}(g)$; set $\text{spans} = \text{false}$. For each $a \in A$:
 $a_idx \leftarrow s.\text{RFIND}(a)$; **if** $a_idx \neq -1$ and $a_idx \leq g_idx$ and $a_idx + \text{LEN}(a) \geq g_end$, set $\text{spans} = \text{true}$. **If** $\neg \text{spans}$, set $\text{alt_present} = \text{false}$.
 5. Return alt_present .
-

model	p_{LR}	AIC_{linear}	AIC_{quad}
DeepSeek-R1-0528	0.000	1032.432	1019.177
DeepSeek-R1-Distill-Llama-70B	0.000	13888.760	13703.165
DeepSeek-R1-Distill-Qwen-1.5B	0.657	11582.864	11584.667
DeepSeek-R1-Distill-Qwen-32B	0.000	16487.488	16409.647
DeepSeek-R1-Distill-Qwen-7B	0.041	14227.155	14224.975
EXAONE-Deep-32B	0.000	15469.929	15388.288
Phi-4-mini-reasoning	0.000	14462.835	14438.087
Phi-4-reasoning	0.000	13344.906	13161.306
Phi-4-reasoning-plus	0.000	12603.062	12477.924
QwQ-32B	0.000	16357.613	16218.181
Qwen3-1.7B	0.000	13700.396	13684.334
Qwen3-30B-A3B	0.000	15363.614	15252.219
Qwen3-32B	0.000	14818.588	14720.097
Qwen3-8B	0.000	15522.392	15425.101
gemini-2.5-flash	0.077	807.553	806.420
gemini-2.5-flash-lite	0.000	1574.306	1555.886
gemini-2.5-pro	0.201	642.222	642.583
gpt-5-2025-08-07	0.001	484.186	474.503
gpt-5-mini-2025-08-07	0.000	1044.689	1023.604
gpt-5-nano-2025-08-07	0.000	1520.921	1500.138
o3-2025-04-16	0.121	692.757	692.356
o4-mini-2025-04-16	0.000	1002.927	970.994

Table 4: Model comparison: linear vs. quadratic fit. A bold number indicates a significant improvement of AIC_{quad} over AIC_{linear} with $p < 0.05$.

Model	d					10
	1	3	5	7	10	
DeepSeek-R1-0528	P400[424]—O4135[6756]	P626[690]—O4697[5531]	P924[1008]—O6194[4777]	P1218[1358]—O6901[8474]	P1218[1358]—O6901[8474]	P1728[1958]—O8225[9710]
DeepSeek-R1-Distill-Llama-70B	P358[404]—O1686[2739]	P616[683]—O2361[3245]	P904[996]—O2456[3364]	P1209[1348]—O2829[3972]	P1209[1348]—O2829[3972]	P1681[1924]—O2473[4067]
DeepSeek-R1-Distill-Qwen-1.5B	P369[415]—O2372[119748]	P627[694]—O2261[64932]	P915[1007]—O20799[61653]	P1220[1359]—O17447[37955]	P1220[1359]—O17447[37955]	P1692[1935]—O17871[36281]
DeepSeek-R1-Distill-Qwen-32B	P369[415]—O2447[3914]	P627[694]—O289[4668]	P915[1007]—O2686[3569]	P1220[1359]—O2776[3908]	P1220[1359]—O2776[3908]	P1692[1935]—O2941[4668]
DeepSeek-R1-Distill-Qwen-7B	P393[438]—O3492[5413]	P627[694]—O4771[6554]	P915[1007]—O5635[7088]	P1220[1359]—O5842[8063]	P1220[1359]—O5842[8063]	P1692[1935]—O4979[6996]
EXAONE-Deep-32B	P369[416]—O7086[11418]	P625[690]—O8012[10967]	P913[1003]—O9423[12726]	P1251[1393]—O11174[14357]	P1251[1393]—O11174[14357]	P1730[1979]—O10971[15880]
Phi-4-mini-reasoning	P579[625]—O3663[5530]	P837[904]—O5182[6575]	P1125[1217]—O6682[8409]	P1430[1569]—O7971[10066]	P1430[1569]—O7971[10066]	P1687[1929]—O9417[12983]
Phi-4-reasoning-plus	P556[625]—O5342[8017]	P823[896]—O6319[8146]	P1124[1218]—O8266[10849]	P1442[1597]—O10442[13677]	P1442[1597]—O10442[13677]	P1902[2145]—O9835[12502]
QwQ-32B	P374[420]—O4194[7729]	P632[699]—O4699[6743]	P920[1012]—O5032[6622]	P1225[1364]—O5239[7108]	P1225[1364]—O5239[7108]	P1697[1940]—O5399[7909]
Qwen3-1.7B	P372[418]—O4774[7658]	P630[697]—O6857[7752]	P918[1010]—O4129[6464]	P1223[1362]—O3030[5337]	P1223[1362]—O3030[5337]	P1695[1938]—O2067[4015]
Qwen3-30B-A3B	P372[418]—O3030[5608]	P630[697]—O4673[6464]	P918[1010]—O4107[5337]	P1223[1362]—O3546[4907]	P1223[1362]—O3546[4907]	P1695[1938]—O3141[5082]
Qwen3-32B	P372[418]—O2307[4211]	P630[697]—O2935[4758]	P918[1010]—O3054[4355]	P1223[1362]—O2902[4322]	P1223[1362]—O2902[4322]	P1695[1938]—O2928[4502]
Qwen3-8B	P372[418]—O4262[6898]	P630[697]—O5222[6698]	P918[1010]—O5048[6469]	P1223[1362]—O4789[6324]	P1223[1362]—O4789[6324]	P1695[1938]—O2967[5742]
gemini-2.5-flash	P364[411]—O2130[2924]	P625[696]—O3947[5396]	P900[997]—O6015[8151]	P1217[1345]—O6926[8780]	P1217[1345]—O6926[8780]	P1699[1935]—O7769[10955]
gemini-2.5-flash-lite	P364[411]—O1073[1246]	P625[696]—O1700[2510]	P900[997]—O1897[2850]	P1217[1345]—O2002[3016]	P1217[1345]—O2002[3016]	P1699[1935]—O1851[3585]
gemini-2.5-pro	P364[411]—O2784[3842]	P625[696]—O4398[6015]	P900[997]—O5895[7953]	P1217[1345]—O7360[9669]	P1217[1345]—O7360[9669]	P1699[1935]—O8311[11803]
gpt-5-2025-08-07	P357[404]—O1228[1750]	P615[685]—O2768[3534]	P889[984]—O3449[4250]	P1207[1334]—O3682[4627]	P1207[1334]—O3682[4627]	P1686[1914]—O4256[5212]
gpt-5-mini-2025-08-07	P357[404]—O1670[2313]	P615[685]—O3066[3555]	P889[984]—O4256[5302]	P1215[1351]—O5182[6320]	P1215[1351]—O5182[6320]	P1686[1914]—O6277[7873]
gpt-5-nano-2025-08-07	P357[404]—O2633[5018]	P615[685]—O4444[6470]	P889[984]—O5917[9075]	P1207[1334]—O6704[9436]	P1207[1334]—O6704[9436]	P1686[1914]—O8127[12956]
o3-2025-04-16	P357[404]—O933[1359]	P615[685]—O2126[2580]	P889[984]—O2685[3213]	P1207[1334]—O2933[3866]	P1207[1334]—O2933[3866]	P1686[1914]—O3571[4685]
o4-mini-2025-04-16	P357[404]—O1682[2462]	P615[685]—O2743[3545]	P889[984]—O3352[4253]	P1207[1334]—O3686[4569]	P1207[1334]—O3686[4569]	P1686[1914]—O4503[5861]

Table 5: Prompt and output tokens by model at N=20. Cells show Pmean[Op]—Omean[Op], Pp=Op=p90.

Model	d					10
	1	3	5	7	10	
DeepSeek-R1-0528	P779[908]—O6060[8073]	P1372[1537]—O7714[9286]	P2021[2195]—O10932[13181]	P2682[2900]—O12144[15058]	P2682[2900]—O12144[15058]	P3658[3946]—O12701[15868]
DeepSeek-R1-Distill-Llama-70B	P774[884]—O2503[3688]	P1362[1500]—O4136[5655]	P1987[2143]—O4427[6201]	P2652[2851]—O4129[6600]	P2652[2851]—O4129[6600]	P3669[3933]—O3245[5604]
DeepSeek-R1-Distill-Qwen-1.5B	P815[925]—O1722[3643]	P1403[1541]—O14514[33405]	P2028[2184]—O13125[32676]	P2693[2892]—O12702[32454]	P2693[2892]—O12702[32454]	P3710[3974]—O14078[32989]
DeepSeek-R1-Distill-Qwen-32B	P815[925]—O3305[5198]	P1403[1541]—O4753[6624]	P2028[2184]—O4671[6376]	P2693[2892]—O4820[7171]	P2693[2892]—O4820[7171]	P3710[3974]—O4162[7217]
DeepSeek-R1-Distill-Qwen-7B	P842[950]—O4721[7396]	P1403[1541]—O6534[9242]	P2028[2184]—O7725[11797]	P2693[2892]—O8318[29548]	P2693[2892]—O8318[29548]	P3710[3974]—O5995[26866]
EXAONE-Deep-32B	P784[896]—O7723[10840]	P1369[1497]—O9218[12190]	P1992[2145]—O9301[13126]	P2654[2848]—O8619[12060]	P2654[2848]—O8619[12060]	P3710[3974]—O9117[13680]
Phi-4-mini-reasoning	P995[1105]—O5733[8056]	P1583[1721]—O8164[10198]	P2208[2364]—O11462[14776]	P2874[3073]—O13659[19401]	P2874[3073]—O13659[19401]	P3890[4154]—O15917[28612]
Phi-4-reasoning-plus	P983[1105]—O7885[11340]	P1596[1745]—O10537[13481]	P2240[2419]—O114875[19947]	P2931[3218]—O19655[29629]	P2931[3218]—O19655[29629]	P4006[4619]—O21226[28904]
QwQ-32B	P820[930]—O5988[9091]	P1408[1546]—O6617[8934]	P2033[2189]—O7580[10243]	P2698[2897]—O7937[11646]	P2698[2897]—O7937[11646]	P3715[3979]—O6928[11473]
Qwen3-1.7B	P818[928]—O4180[7015]	P1406[1544]—O5482[7856]	P2031[2187]—O3682[6167]	P2696[2895]—O2492[4879]	P2696[2895]—O2492[4879]	P3713[3977]—O1867[4431]
Qwen3-30B-A3B	P818[928]—O3847[5532]	P1406[1544]—O5600[7227]	P2031[2187]—O5240[6894]	P2696[2895]—O5077[7280]	P2696[2895]—O5077[7280]	P3713[3977]—O4451[7365]
Qwen3-32B	P818[928]—O3547[5774]	P1406[1544]—O4416[6680]	P2031[2187]—O4189[6276]	P2696[2895]—O4045[6100]	P2696[2895]—O4045[6100]	P3713[3977]—O3730[6143]
Qwen3-8B	P818[928]—O5183[7561]	P1406[1544]—O6971[8185]	P2031[2187]—O6023[8164]	P2696[2895]—O5784[7660]	P2696[2895]—O5784[7660]	P3713[3977]—O4879[7485]
gemini-2.5-flash	P810[922]—O4343[6115]	P1373[1501]—O7708[10836]	P2030[2208]—O12945[17344]	P2688[2883]—O16780[22526]	P2688[2883]—O16780[22526]	P3691[3957]—O18449[28927]
gemini-2.5-flash-lite	P810[922]—O1840[2739]	P1373[1501]—O2971[5314]	P2030[2208]—O3570[5976]	P2688[2883]—O2867[5504]	P2688[2883]—O2867[5504]	P3691[3957]—O1510[5970]
gemini-2.5-pro	P810[922]—O5958[8294]	P1373[1501]—O9435[13393]	P2030[2208]—O15287[17418]	P2688[2883]—O14723[20703]	P2688[2883]—O14723[20703]	P3691[3957]—O16236[21429]
gpt-5-2025-08-07	P770[884]—O2112[2900]	P1335[1463]—O5251[6805]	P1992[2169]—O5976[7904]	P2644[2830]—O6239[8467]	P2644[2830]—O6239[8467]	P3646[3914]—O6534[8732]
gpt-5-mini-2025-08-07	P770[884]—O2670[3788]	P1335[1463]—O5794[6779]	P1992[2169]—O8959[11716]	P2644[2830]—O10562[14197]	P2644[2830]—O10562[14197]	P3646[3914]—O9879[15575]
gpt-5-nano-2025-08-07	P770[884]—O4146[6810]	P1335[1463]—O7488[11220]	P1992[2169]—O10491[15707]	P2644[2830]—O10557[16212]	P2644[2830]—O10557[16212]	P3646[3914]—O9268[16136]
o3-2025-04-16	P770[884]—O1704[2388]	P1335[1463]—O4123[5006]	P1992[2169]—O5237[7181]	P2644[2830]—O3571[8103]	P2644[2830]—O3571[8103]	P3646[3914]—O5411[7861]
o4-mini-2025-04-16	P770[884]—O2692[4426]	P1335[1463]—O4439[5671]	P1992[2169]—O6120[9511]	P2644[2830]—O5831[9118]	P2644[2830]—O5831[9118]	P3646[3914]—O6446[9573]

Table 6: Prompt and output tokens by model at N=50. Cells show Pmean[Op]—Omean[Op], Pp=Op=p90.

Model	d					10			
	1	3	5	7	10				
DeepSeek-R1-0528	P1457[1684]	O8142[11134]	P2627[2930]	P3873[4119]	O14319[18636]	P5035[5376]	O11861[17026]	P6860[7253]	O10608[13931]
DeepSeek-R1-Distill-Llama-70B	P1468[1684]	O3628[5573]	P2630[2904]	P3793[4071]	O6587[10134]	P5009[5308]	O3706[10984]	P6878[7239]	O4133[8443]
DeepSeek-R1-Distill-Qwen-1.5B	P1560[1776]	O16989[34188]	P2722[2996]	P3885[4163]	O10608[31366]	P5101[5400]	O10769[29845]	P6970[7331]	O9680[28165]
DeepSeek-R1-Distill-Qwen-32B	P1560[1776]	O4480[7023]	P2722[2996]	P3885[4163]	O7847[11083]	P5101[5400]	O7047[11282]	P6970[7331]	O5578[10228]
DeepSeek-R1-Distill-Qwen-7B	P1560[1776]	O5504[9157]	P2722[2996]	P3885[4163]	O10153[32087]	P5101[5400]	O8469[31107]	P6970[7331]	O8149[30381]
EXAONE-Deep-32B	P1588[1859]	O13168[17829]	P2758[3051]	P3931[4219]	O13399[18030]	P5160[5470]	O10195[14815]	P7034[7399]	O7605[11342]
Phi-4-mini-reasoning	P1475[1695]	O8778[12190]	P2633[2908]	P3792[4060]	O8457[12413]	P5003[5302]	O7324[10152]	P6864[7220]	O6699[9230]
Phi-4-reasoning	P1689[1905]	O8360[11469]	P2851[3125]	P4015[4295]	O15167[22715]	P5230[5529]	O16521[27514]	P7103[7462]	O16996[25674]
Phi-4-reasoning-plus	P1694[1968]	O12270[16909]	P2910[3238]	P4123[4586]	O20288[28560]	P5309[5620]	O19976[27662]	P7309[8426]	O19392[25782]
QwQ-32B	P1565[1781]	O7625[10956]	P2727[3001]	P3890[4168]	O9089[13166]	P5106[5405]	O9096[13950]	P6975[7336]	O8298[12846]
Qwen3-1.7B	P1563[1779]	O4079[7309]	P2725[2999]	P3888[4166]	O4191[7665]	P5104[5403]	O2933[6423]	P6973[7334]	O2144[7498]
Qwen3-30B-A3B	P1563[1779]	O4949[7052]	P2725[2999]	P3888[4166]	O6443[9249]	P5104[5403]	O5233[8618]	P6973[7334]	O4263[8469]
Qwen3-32B	P1563[1779]	O5141[7983]	P2725[2999]	P3888[4166]	O5157[7311]	P5104[5403]	O4638[7052]	P6973[7334]	O3961[6269]
Qwen3-8B	P1563[1779]	O6076[8098]	P2725[2999]	P3888[4166]	O7797[10327]	P5104[5403]	O7076[10829]	P6973[7334]	O5596[10241]
gemini-2.5-flash	P1536[1771]	O8290[13046]	P2699[2990]	P3910[4123]	O25507[29050]	P5087[5439]	O25224[28010]	P6909[7279]	O23637[26097]
gemini-2.5-flash-lite	P1536[1771]	O2526[3917]	P2699[2990]	P3910[4123]	O5391[10345]	P5087[5439]	O2846[8615]	P6909[7279]	O1612[7689]
gemini-2.5-pro	P1536[1771]	O10927[15454]	P2699[2990]	P3910[4123]	O21258[28842]	P5087[5439]	O21546[27761]	P6909[7279]	O22577[26042]
gpt-5-2025-08-07	P1446[1683]	O3294[4693]	P2606[2884]	P3814[4016]	O8165[11989]	P4992[5333]	O8733[13057]	P6808[7173]	O9283[12365]
gpt-5-mini-2025-08-07	P1446[1683]	O3808[5621]	P2606[2884]	P3814[4016]	O13104[18082]	P5037[5337]	O12790[21517]	P6808[7173]	O12440[17742]
gpt-5-nano-2025-08-07	P1607[1683]	O5999[9006]	P2627[2905]	P3814[4016]	O13450[19789]	P4992[5333]	O9336[17104]	P6808[7173]	O8322[15122]
o3-2025-04-16	P1446[1683]	O2717[4066]	P2606[2884]	P3814[4016]	O6959[10925]	P4992[5333]	O6897[10914]	P6808[7173]	O8748[14425]
o4-mini-2025-04-16	P1446[1683]	O4153[6167]	P2606[2884]	P3814[4016]	O7910[13477]	P4992[5333]	O7269[12320]	P6808[7173]	O8365[16052]

Table 7: Prompt and output tokens by model at N=100. Cells show Pmean[Op], Pp=Op=p90.

Model	d					10			
	1	3	5	7	10				
DeepSeek-R1-0528	P3639[4255]	O11660[15547]	P6442[6992]	P9132[9696]	O10608[13547]	P12014[12793]	O10417[13038]	P16345[17021]	O9524[11707]
DeepSeek-R1-Distill-Llama-70B	P3547[4084]	O4738[7376]	P6355[6974]	P9173[9789]	O8206[16317]	P12024[12660]	O6814[15722]	P16395[16998]	O4547[11339]
DeepSeek-R1-Distill-Qwen-1.5B	P3939[4476]	O16892[32864]	P6747[7366]	P9565[10181]	O11895[28483]	P12416[13052]	O11754[28378]	P16787[17390]	O8652[19503]
DeepSeek-R1-Distill-Qwen-32B	P3939[4476]	O5778[10068]	P6747[7366]	P9565[10181]	O15297[23500]	P12416[13052]	O12679[22803]	P16787[17390]	O9262[21900]
DeepSeek-R1-Distill-Qwen-7B	P3939[4476]	O5362[10426]	P6747[7366]	P9565[10181]	O8579[26492]	P12416[13052]	O8531[25679]	P16787[17390]	O7354[23761]
EXAONE-Deep-32B	P3977[4500]	O12691[16997]	P6801[7452]	P9641[10282]	O11156[15514]	P12513[13177]	O8395[12329]	P16933[17577]	O5523[9281]
Phi-4-mini-reasoning	P3550[4096]	O8854[12622]	P6345[6934]	P9153[9758]	O7091[10159]	P11993[12610]	O6053[8649]	P16345[16948]	O5462[7771]
Phi-4-reasoning	P3768[4305]	O14056[19926]	P6576[7195]	P9394[10010]	O18249[23605]	P12245[12881]	O18222[20994]	P16616[17219]	O14578[16740]
Phi-4-reasoning-plus	P3830[4315]	O20039[28464]	P6796[7802]	P9739[11372]	O20595[23689]	P12670[14810]	O19383[21013]	P17347[20890]	O15076[16718]
QwQ-32B	P3944[4481]	O9353[13906]	P6752[7371]	P9570[10186]	O13878[21556]	P12421[13057]	O12844[20036]	P16792[17395]	O10085[16212]
Qwen3-1.7B	P3942[4479]	O3919[7534]	P6750[7369]	P9568[10184]	O5147[12851]	P12419[13055]	O4673[10236]	P16792[17394]	O2718[13976]
Qwen3-30B-A3B	P3942[4479]	O7036[10502]	P6750[7369]	P9568[10184]	O7500[14503]	P12419[13055]	O5664[13342]	P16790[17393]	O3418[8000]
Qwen3-32B	P3942[4479]	O6840[9827]	P6750[7369]	P9568[10184]	O6474[9809]	P12419[13055]	O5034[8153]	P16790[17393]	O3654[6255]
Qwen3-8B	P3942[4479]	O8146[12378]	P6750[7369]	P9568[10184]	O9710[18564]	P12419[13055]	O8019[17663]	P16790[17393]	O5349[13378]
gemini-2.5-flash	P3995[4683]	O5843[10440]	P6768[7256]	P9474[10084]	O23629[24363]	P12351[13131]	O20829[21494]	P16683[17372]	O16508[17088]
gemini-2.5-flash-lite	P3995[4683]	O21432[28925]	P6768[7256]	P9474[10084]	O6728[21344]	P12351[13131]	O3665[19711]	P16683[17372]	O3666[9]
gemini-2.5-pro	P3596[4232]	O5910[7532]	P6376[6874]	P9076[9672]	O13513[17813]	P11944[12654]	O20327[21494]	P16265[16934]	O13799[16902]
gpt-5-2025-08-07	P3596[4232]	O6472[8818]	P6376[6874]	P9076[9672]	O15604[20850]	P11944[12654]	O15981[20713]	P16274[16934]	O13598[16832]
gpt-5-mini-2025-08-07	P3762[4246]	O10611[16059]	P6376[6874]	P9076[9672]	O8661[17818]	P11944[12654]	O6539[16148]	P16267[16934]	O4498[11784]
gpt-5-nano-2025-08-07	P3589[4232]	O4971[7131]	P6376[6874]	P9076[9672]	O10748[16757]	P11944[12654]	O12022[19860]	P16265[16934]	O11833[16704]
o3-2025-04-16	P3604[4232]	O6679[9945]	P6376[6874]	P9076[9672]	O11155[19673]	P11948[12662]	O10845[19674]	P16265[16934]	O10263[16382]

Table 8: Prompt and output tokens by model at N=250. Cells show Pmean[Op], Pp=Op=p90.